```
        1996     1996#    1998     1998#    2000     2000#    2002     2002#    2004     2004#    2007     2007#    2009     2009#    SEQ 1249
        2011     2011#    2013     2013#    2015     2015#    2017     2017#    2019     2019#    2021     2021#    2023     2023#
        2025     2025#    2027     2027#    2029     2029#    2031     2031#    2033     2033#    2035     2035#    2037     2037#
        2040     2040#    2042     2042#    2044     2044#    2046     2046#    2048     2048#    2050     2050#    2052     2052#
        2054     2054#    2056     2056#    2058     2058#    2060     2060#    2062     2062#    2064     2064#    2066     2066#
        2068     2068#    2070     2070#    2073     2073#    2075     2075#    2077     2077#    2079     2079#    2081     2081#
        2083     2083#    2085     2085#    2087     2087#    2089     2089#    2091     2091#    2093     2093#    2095     2095#
        2097     2097#    2099     2099#    2101     2101#    2103     2103#    2108     2108#    2110     2110#    2113     2113#
        2115     2115#    2117     2117#    2119     2119#    2121     2121#    2123     2123#    2125     2125#    2127     2127#
        2129     2129#    2131     2131#    2133     2133#    2135     2135#    2137     2137#    2139     2139#    2141     2141#
        2143     2143#    2146     2146#    2148     2148#    2150     2150#    2152     2152#    2154     2154#    2156     2156#
        2158     2158#    2160     2160#    2162     2162#    2164     2164#    2166     2166#    2168     2168#    2170     2170#
        2172     2172#    2174     2174#    2176     2176#    2179     2179#    2181     2181#    2183     2183#    2185     2185#
        2187     2187#    2189     2189#    2191     2191#    2193     2193#    2195     2195#    2197     2197#    2199     2199#
        2201     2201#    2203     2203#    2205     2205#    2207     2207#    2209     2209#    2212     2212#    2214     2214#
        2216     2216#    2218     2218#    2220     2220#    2222     2222#    2224     2224#    2226     2226#    2228     2228#
        2230     2230#    2232     2232#    2234     2234#    2236     2236#    2238     2238#    2240     2240#    2242     2242#
        2247     2247#    2249     2249#    2252     2252#    2254     2254#    2256     2256#    2258     2258#    2260     2260#
        2262     2262#    2264     2264#    2266     2266#    2268     2268#    2270     2270#    2272     2272#    2274     2274#
        2276     2276#    2278     2278#    2280     2280#    2282     2282#    2285     2285#    2287     2287#    2289     2289#
        2291     2291#    2293     2293#    2295     2295#    2297     2297#    2299     2299#    2301     2301#    2303     2303#
        2305     2305#    2307     2307#    2309     2309#    2311     2311#    2313     2313#    2315     2315#    2318     2318#
        2320     2320#    2322     2322#    2324     2324#    2326     2326#    2328     2328#    2330     2330#    2332     2332#
        2334     2334#    2336     2336#    2338     2338#    2340     2340#    2342     2342#    2344     2344#    2346     2346#
        2348     2348#    2351     2351#    2353     2353#    2355     2355#    2357     2357#    2359     2359#    2361     2361#
        2363     2363#    2365     2365#    2367     2367#    2369     2369#    2371     2371#    2373     2373#    2375     2375#
        2377     2377#    2379     2379#    2381     2381#    2386     2386#    2388     2388#    2391     2391#    2393     2393#
        2395     2395#    2397     2397#    2399     2399#    2401     2401#    2403     2403#    2405     2405#    2407     2407#
        2409     2409#    2411     2411#    2413     2413#    2415     2415#    2417     2417#    2419     2419#    2421     2421#
        2424     2424#    2426     2426#    2428     2428#    2430     2430#    2432     2432#    2434     2434#    2436     2436#
        2438     2438#    2440     2440#    2442     2442#    2444     2444#    2446     2446#    2448     2448#    2450     2450#
        2452     2452#    2454     2454#    2457     2457#    2459     2459#    2461     2461#    2463     2463#    2465     2465#
        2467     2467#    2469     2469#    2471     2471#    2473     2473#    2475     2475#    2477     2477#    2479     2479#
        2481     2481#    2483     2483#    2485     2485#    2487     2487#    2490     2490#    2492     2492#    2494     2494#
        2496     2496#    2498     2498#    2500     2500#    2502     2502#    2504     2504#    2506     2506#    2508     2508#
        2510     2510#    2512     2512#    2514     2514#    2516     2516#    2518     2518#    2520     2520#    2522     2522#
%ML        103#     103       105#     105       111#     111       113#     113       115#     115       117#     117       119#     119
           121#     121       123#     123       125#     125       127#     127       129#     129       131#     131       133#     133
           135#     135       137#     137       139#     139       141#     141       143#     143       145#     145       150#     150
           152#     152       154#     154       159#     159       161#     161       163#     163       168#     168       170#     170
           172#     172       177#     177       179#     179       181#     181       186#     186       188#     188       190#     190
           195#     195       197#     197       199#     199       204#     204       206#     206       208#     208       213#     213
           215#     215       217#     217       222#     222       224#     224       226#     226       231#     231       233#     233
           235#     235       240#     240       242#     242       244#     244       249#     249       251#     251       253#     253
           258#     258       260#     260       262#     262       267#     267       269#     269       271#     271       276#     276
           278#     278       280#     280       285#     285       287#     287       289#     289       294#     294       296#     296
           301#     301       303#     303       306#     306       308#     308       310#     310       312#     312       314#     314
           316#     316       318#     318       320#     320       322#     322       324#     324       326#     326       328#     328
           330#     330       332#     332       334#     334       336#     336       339#     339       341#     341       343#     343
           345#     345       347#     347       349#     349       351#     351       353#     353       355#     355       357#     357
           359#     359       361#     361       363#     363       365#     365       367#     367       369#     369       372#     372
           374#     374       376#     376       378#     378       380#     380       382#     382       384#     384       386#     386
           388#     388       390#     390       392#     392       394#     394       396#     396       398#     398       400#     400
```

```
 402#   402    405#   405    407#   407    409#   409    411#   411    413#   413    415#   415     SEQ 1250
 417#   417    419#   419    421#   421    423#   423    425#   425    427#   427    429#   429
 431#   431    433#   433    435#   435    440#   440    442#   442    445#   445    447#   447
 449#   449    451#   451    453#   453    455#   455    457#   457    459#   459    461#   461
 463#   463    465#   465    467#   467    469#   469    471#   471    473#   473    475#   475
 478#   478    480#   480    482#   482    484#   484    486#   486    488#   488    490#   490
 492#   492    494#   494    496#   496    498#   498    500#   500    502#   502    504#   504
 506#   506    508#   508    511#   511    513#   513    515#   515    517#   517    519#   519
 521#   521    523#   523    525#   525    527#   527    529#   529    531#   531    533#   533
 535#   535    537#   537    539#   539    541#   541    544#   544    546#   546    548#   548
 550#   550    552#   552    554#   554    556#   556    558#   558    560#   560    562#   562
 564#   564    566#   566    568#   568    570#   570    572#   572    574#   574    579#   579
 581#   581    584#   584    586#   586    588#   588    590#   590    592#   592    594#   594
 596#   596    598#   598    600#   600    602#   602    604#   604    606#   606    608#   608
 610#   610    612#   612    614#   614    617#   617    619#   619    621#   621    623#   623
 625#   625    627#   627    629#   629    631#   631    633#   633    635#   635    637#   637
 639#   639    641#   641    643#   643    645#   645    647#   647    650#   650    652#   652
 654#   654    656#   656    658#   658    660#   660    662#   662    664#   664    666#   666
 668#   668    670#   670    672#   672    674#   674    676#   676    678#   678    680#   680
 683#   683    685#   685    687#   687    689#   689    691#   691    693#   693    695#   695
 697#   697    699#   699    701#   701    703#   703    705#   705    707#   707    709#   709
 711#   711    713#   713    718#   718    720#   720    723#   723    725#   725    727#   727
 729#   729    731#   731    733#   733    735#   735    737#   737    739#   739    741#   741
 743#   743    745#   745    747#   747    749#   749    751#   751    753#   753    756#   756
 758#   758    760#   760    762#   762    764#   764    766#   766    768#   768    770#   770
 772#   772    774#   774    776#   776    778#   778    780#   780    782#   782    784#   784
 786#   786    789#   789    791#   791    793#   793    795#   795    797#   797    799#   799
 801#   801    803#   803    805#   805    807#   807    809#   809    811#   811    813#   813
 815#   815    817#   817    819#   819    822#   822    824#   824    826#   826    828#   828
 830#   830    832#   832    834#   834    836#   836    838#   838    840#   840    842#   842
 844#   844    846#   846    848#   848    850#   850    852#   852    857#   857    859#   859
 862#   862    864#   864    866#   866    868#   868    870#   870    872#   872    874#   874
 876#   876    878#   878    880#   880    882#   882    884#   884    886#   886    888#   888
 890#   890    892#   892    895#   895    897#   897    899#   899    901#   901    903#   903
 905#   905    907#   907    909#   909    911#   911    913#   913    915#   915    917#   917
 919#   919    921#   921    923#   923    925#   925    928#   928    930#   930    932#   932
 934#   934    936#   936    938#   938    940#   940    942#   942    944#   944    946#   946
 948#   948    950#   950    952#   952    954#   954    956#   956    958#   958    961#   961
 963#   963    965#   965    967#   967    969#   969    971#   971    973#   973    975#   975
 977#   977    979#   979    981#   981    983#   983    985#   985    987#   987    989#   989
 991#   991    996#   996    998#   998   1001#  1001   1003#  1003   1005#  1005   1007#  1007
1009#  1009   1011#  1011   1013#  1013   1015#  1015   1017#  1017   1019#  1019   1021#  1021
1023#  1023   1025#  1025   1027#  1027   1029#  1029   1031#  1031   1034#  1034   1036#  1036
1038#  1038   1040#  1040   1042#  1042   1044#  1044   1046#  1046   1048#  1048   1050#  1050
1052#  1052   1054#  1054   1056#  1056   1058#  1058   1060#  1060   1062#  1062   1064#  1064
1067#  1067   1069#  1069   1071#  1071   1073#  1073   1075#  1075   1077#  1077   1079#  1079
1081#  1081   1083#  1083   1085#  1085   1087#  1087   1089#  1089   1091#  1091   1093#  1093
1095#  1095   1097#  1097   1100#  1100   1102#  1102   1104#  1104   1106#  1106   1108#  1108
1110#  1110   1112#  1112   1114#  1114   1116#  1116   1118#  1118   1120#  1120   1122#  1122
1124#  1124   1126#  1126   1128#  1128   1130#  1130   1135#  1135   1137#  1137   1140#  1140
1142#  1142   1144#  1144   1146#  1146   1148#  1148   1150#  1150   1152#  1152   1154#  1154
1156#  1156   1158#  1158   1160#  1160   1162#  1162   1164#  1164   1166#  1166   1168#  1168
1170#  1170   1173#  1173   1175#  1175   1177#  1177   1179#  1179   1181#  1181   1183#  1183
```

```
1185#   1185    1187#   1187    1189#   1189    1191#   1191    1193#   1193    1195#   1195    1197#   1197     SEQ 1251
1199#   1199    1201#   1201    1203#   1203    1206#   1206    1208#   1208    1210#   1210    1212#   1212
1214#   1214    1216#   1216    1218#   1218    1220#   1220    1222#   1222    1224#   1224    1226#   1226
1228#   1228    1230#   1230    1232#   1232    1234#   1234    1236#   1236    1239#   1239    1241#   1241
1243#   1243    1245#   1245    1247#   1247    1249#   1249    1251#   1251    1253#   1253    1255#   1255
1257#   1257    1259#   1259    1261#   1261    1263#   1263    1265#   1265    1267#   1267    1269#   1269
1274#   1274    1276#   1276    1279#   1279    1281#   1281    1283#   1283    1285#   1285    1287#   1287
1289#   1289    1291#   1291    1293#   1293    1295#   1295    1297#   1297    1299#   1299    1301#   1301
1303#   1303    1305#   1305    1307#   1307    1309#   1309    1312#   1312    1314#   1314    1316#   1316
1318#   1318    1320#   1320    1322#   1322    1324#   1324    1326#   1326    1328#   1328    1330#   1330
1332#   1332    1334#   1334    1336#   1336    1338#   1338    1340#   1340    1342#   1342    1345#   1345
1347#   1347    1349#   1349    1351#   1351    1353#   1353    1355#   1355    1357#   1357    1359#   1359
1361#   1361    1363#   1363    1365#   1365    1367#   1367    1369#   1369    1371#   1371    1373#   1373
1375#   1375    1378#   1378    1380#   1380    1382#   1382    1384#   1384    1386#   1386    1388#   1388
1390#   1390    1392#   1392    1394#   1394    1396#   1396    1398#   1398    1400#   1400    1402#   1402
1404#   1404    1406#   1406    1408#   1408    1413#   1413    1415#   1415    1418#   1418    1420#   1420
1422#   1422    1424#   1424    1426#   1426    1428#   1428    1430#   1430    1432#   1432    1434#   1434
1436#   1436    1438#   1438    1440#   1440    1442#   1442    1444#   1444    1446#   1446    1448#   1448
1451#   1451    1453#   1453    1455#   1455    1457#   1457    1459#   1459    1461#   1461    1463#   1463
1465#   1465    1467#   1467    1469#   1469    1471#   1471    1473#   1473    1475#   1475    1477#   1477
1479#   1479    1481#   1481    1484#   1484    1486#   1486    1488#   1488    1490#   1490    1492#   1492
1494#   1494    1496#   1496    1498#   1498    1500#   1500    1502#   1502    1504#   1504    1506#   1506
1508#   1508    1510#   1510    1512#   1512    1514#   1514    1517#   1517    1519#   1519    1521#   1521
1523#   1523    1525#   1525    1527#   1527    1529#   1529    1531#   1531    1533#   1533    1535#   1535
1537#   1537    1539#   1539    1541#   1541    1543#   1543    1545#   1545    1547#   1547    1552#   1552
1554#   1554    1557#   1557    1559#   1559    1561#   1561    1563#   1563    1565#   1565    1567#   1567
1569#   1569    1571#   1571    1573#   1573    1575#   1575    1577#   1577    1579#   1579    1581#   1581
1583#   1583    1585#   1585    1587#   1587    1590#   1590    1592#   1592    1594#   1594    1596#   1596
1598#   1598    1600#   1600    1602#   1602    1604#   1604    1606#   1606    1608#   1608    1610#   1610
1612#   1612    1614#   1614    1616#   1616    1618#   1618    1620#   1620    1623#   1623    1625#   1625
1627#   1627    1629#   1629    1631#   1631    1633#   1633    1635#   1635    1637#   1637    1639#   1639
1641#   1641    1643#   1643    1645#   1645    1647#   1647    1649#   1649    1651#   1651    1653#   1653
1656#   1656    1658#   1658    1660#   1660    1662#   1662    1664#   1664    1666#   1666    1668#   1668
1670#   1670    1672#   1672    1674#   1674    1676#   1676    1678#   1678    1680#   1680    1682#   1682
1684#   1684    1686#   1686    1691#   1691    1693#   1693    1696#   1696    1698#   1698    1700#   1700
1702#   1702    1704#   1704    1706#   1706    1708#   1708    1710#   1710    1712#   1712    1714#   1714
1716#   1716    1718#   1718    1720#   1720    1722#   1722    1724#   1724    1726#   1726    1729#   1729
1731#   1731    1733#   1733    1735#   1735    1737#   1737    1739#   1739    1741#   1741    1743#   1743
1745#   1745    1747#   1747    1749#   1749    1751#   1751    1753#   1753    1755#   1755    1757#   1757
1759#   1759    1762#   1762    1764#   1764    1766#   1766    1768#   1768    1770#   1770    1772#   1772
1774#   1774    1776#   1776    1778#   1778    1780#   1780    1782#   1782    1784#   1784    1786#   1786
1788#   1788    1790#   1790    1792#   1792    1795#   1795    1797#   1797    1799#   1799    1801#   1801
1803#   1803    1805#   1805    1807#   1807    1809#   1809    1811#   1811    1813#   1813    1815#   1815
1817#   1817    1819#   1819    1821#   1821    1823#   1823    1825#   1825    1830#   1830    1832#   1832
1835#   1835    1837#   1837    1839#   1839    1841#   1841    1843#   1843    1845#   1845    1847#   1847
1849#   1849    1851#   1851    1853#   1853    1855#   1855    1857#   1857    1859#   1859    1861#   1861
1863#   1863    1865#   1865    1868#   1868    1870#   1870    1872#   1872    1874#   1874    1876#   1676
1878#   1878    1880#   1880    1882#   1882    1884#   1884    1886#   1886    1888#   1888    1890#   1890
1892#   1892    1894#   1894    1896#   1896    1898#   1898    1901#   1901    1903#   1903    1905#   1905
1907#   1907    1909#   1909    1911#   1911    1913#   1913    1915#   1915    1917#   1917    1919#   1919
1921#   1921    1923#   1923    1925#   1925    1927#   1927    1929#   1929    1931#   1931    1934#   1934
1936#   1936    1938#   1938    1940#   1940    1942#   1942    1944#   1944    1946#   1946    1948#   1948
1950#   1950    1952#   1952    1954#   1954    1956#   1956    1958#   1958    1960#   1960    1962#   1962
```

```
      1964#  1964    1969#  1969    1971#  1971    1974#  1974    1976#  1976    1978#  1978    1980#  1980    SEQ 1252
      1982#  1982    1984#  1984    1986#  1986    1988#  1988    1990#  1990    1992#  1992    1994#  1994
      1996#  1996    1998#  1998    2000#  2000    2002#  2002    2004#  2004    2007#  2007    2009#  2009
      2011#  2011    2013#  2013    2015#  2015    2017#  2017    2019#  2019    2021#  2021    2023#  2023
      2025#  2025    2027#  2027    2029#  2029    2031#  2031    2033#  2033    2035#  2035    2037#  2037
      2040#  2040    2042#  2042    2044#  2044    2046#  2046    2048#  2048    2050#  2050    2052#  2052
      2054#  2054    2056#  2056    2058#  2058    2060#  2060    2062#  2062    2064#  2064    2066#  2066
      2068#  2068    2070#  2070    2073#  2073    2075#  2075    2077#  2077    2079#  2079    2081#  2081
      2083#  2083    2085#  2085    2087#  2087    2089#  2089    2091#  2091    2093#  2093    2095#  2095
      2097#  2097    2099#  2099    2101#  2101    2103#  2103    2108#  2108    2110#  2110    2113#  2113
      2115#  2115    2117#  2117    2119#  2119    2121#  2121    2123#  2123    2125#  2125    2127#  2127
      2129#  2129    2131#  2131    2133#  2133    2135#  2135    2137#  2137    2139#  2139    2141#  2141
      2143#  2143    2146#  2146    2148#  2148    2150#  2150    2152#  2152    2154#  2154    2156#  2156
      2158#  2158    2160#  2160    2162#  2162    2164#  2164    2166#  2166    2168#  2168    2170#  2170
      2172#  2172    2174#  2174    2176#  2176    2179#  2179    2181#  2181    2183#  2183    2185#  2185
      2187#  2187    2189#  2189    2191#  2191    2193#  2193    2195#  2195    2197#  2197    2199#  2199
      2201#  2201    2203#  2203    2205#  2205    2207#  2207    2209#  2209    2212#  2212    2214#  2214
      2216#  2216    2218#  2218    2220#  2220    2222#  2222    2224#  2224    2226#  2226    2228#  2228
      2230#  2230    2232#  2232    2234#  2234    2236#  2236    2238#  2238    2240#  2240    2242#  2242
      2247#  2247    2249#  2249    2252#  2252    2254#  2254    2256#  2256    2258#  2258    2260#  2260
      2262#  2262    2264#  2264    2266#  2266    2268#  2268    2270#  2270    2272#  2272    2274#  2274
      2276#  2276    2278#  2278    2280#  2280    2282#  2282    2285#  2285    2287#  2287    2289#  2289
      2291#  2291    2293#  2293    2295#  2295    2297#  2297    2299#  2299    2301#  2301    2303#  2303
      2305#  2305    2307#  2307    2309#  2309    2311#  2311    2313#  2313    2315#  2315    2318#  2318
      2320#  2320    2322#  2322    2324#  2324    2326#  2326    2328#  2328    2330#  2330    2332#  2332
      2334#  2334    2336#  2336    2338#  2338    2340#  2340    2342#  2342    2344#  2344    2346#  2346
      2348#  2348    2351#  2351    2353#  2353    2355#  2355    2357#  2357    2359#  2359    2361#  2361
      2363#  2363    2365#  2365    2367#  2367    2369#  2369    2371#  2371    2373#  2373    2375#  2375
      2377#  2377    2379#  2379    2381#  2381    2386#  2386    2388#  2388    2391#  2391    2393#  2393
      2395#  2395    2397#  2397    2399#  2399    2401#  2401    2403#  2403    2405#  2405    2407#  2407
      2409#  2409    2411#  2411    2413#  2413    2415#  2415    2417#  2417    2419#  2419    2421#  2421
      2424#  2424    2426#  2426    2428#  2428    2430#  2430    2432#  2432    2434#  2434    2436#  2436
      2438#  2438    2440#  2440    2442#  2442    2444#  2444    2446#  2446    2448#  2448    2450#  2450
      2452#  2452    2454#  2454    2457#  2457    2459#  2459    2461#  2461    2463#  2463    2465#  2465
      2467#  2467    2469#  2469    2471#  2471    2473#  2473    2475#  2475    2477#  2477    2479#  2479
      2481#  2481    2483#  2483    2485#  2485    2487#  2487    2490#  2490    2492#  2492    2494#  2494
      2496#  2496    2498#  2498    2500#  2500    2502#  2502    2504#  2504    2506#  2506    2508#  2508
      2510#  2510    2512#  2512    2514#  2514    2516#  2516    2518#  2518    2520#  2520    2522#  2522
%MR    103#   103     105#   105     111#   111     113#   113     115#   115     117#   117     119#   119
       121#   121     123#   123     125#   125     127#   127     129#   129     131#   131     133#   133
       135#   135     137#   137     139#   139     141#   141     143#   143     145#   145     150#   150
       152#   152     154#   154     159#   159     161#   161     163#   163     168#   168     170#   170
       172#   172     177#   177     179#   179     181#   181     186#   186     188#   188     190#   190
       195#   195     197#   197     199#   199     204#   204     206#   206     208#   208     213#   213
       215#   215     217#   217     222#   222     224#   224     226#   226     231#   231     233#   233
       235#   235     240#   240     242#   242     244#   244     249#   249     251#   251     253#   253
       258#   258     260#   260     262#   262     267#   267     269#   269     271#   271     276#   276
       278#   278     280#   280     285#   285     287#   287     289#   289     294#   294     296#   296
       301#   301     303#   303     306#   306     308#   308     310#   310     312#   312     314#   314
       316#   316     318#   318     320#   320     322#   322     324#   324     326#   326     328#   328
       330#   330     332#   332     334#   334     336#   336     339#   339     341#   341     343#   343
       345#   345     347#   347     349#   349     351#   351     353#   353     355#   355     357#   357
       359#   359     361#   361     363#   363     365#   365     367#   367     369#   369     372#   372
```

```
 374#   374     376#   376     378#   378     380#   380     382#   382     384#   384     386#   386    SEQ 1253
 388#   388     390#   390     392#   392     394#   394     396#   396     398#   398     400#   400
 402#   402     405#   405     407#   407     409#   409     411#   411     413#   413     415#   415
 417#   417     419#   419     421#   421     423#   423     425#   425     427#   427     429#   429
 431#   431     433#   433     435#   435     440#   440     442#   442     445#   445     447#   447
 449#   449     451#   451     453#   453     455#   455     457#   457     459#   459     461#   461
 463#   463     465#   465     467#   467     469#   469     471#   471     473#   473     475#   475
 478#   478     480#   480     482#   482     484#   484     486#   486     488#   488     490#   490
 492#   492     494#   494     496#   496     498#   498     500#   500     502#   502     504#   504
 506#   506     508#   508     511#   511     513#   513     515#   515     517#   517     519#   519
 521#   521     523#   523     525#   525     527#   527     529#   529     531#   531     533#   533
 535#   535     537#   537     539#   539     541#   541     544#   544     546#   546     548#   548
 550#   550     552#   552     554#   554     556#   556     558#   558     560#   560     562#   562
 564#   564     566#   566     568#   568     570#   570     572#   572     574#   574     579#   579
 581#   581     584#   584     586#   586     588#   588     590#   590     592#   592     594#   594
 596#   596     598#   598     600#   600     602#   602     604#   604     606#   606     608#   608
 610#   610     612#   612     614#   614     617#   617     619#   619     621#   621     623#   623
 625#   625     627#   627     629#   629     631#   631     633#   633     635#   635     637#   637
 639#   639     641#   641     643#   643     645#   645     647#   647     650#   650     652#   652
 654#   654     656#   656     658#   658     660#   660     662#   662     664#   664     666#   666
 668#   668     670#   670     672#   672     674#   674     676#   676     678#   678     680#   680
 683#   683     685#   685     687#   687     689#   689     691#   691     693#   693     695#   695
 697#   697     699#   699     701#   701     703#   703     705#   705     707#   707     709#   709
 711#   711     713#   713     718#   718     720#   720     723#   723     725#   725     727#   727
 729#   729     731#   731     733#   733     735#   735     737#   737     739#   739     741#   741
 743#   743     745#   745     747#   747     749#   749     751#   751     753#   753     756#   756
 758#   758     760#   760     762#   762     764#   764     766#   766     768#   768     770#   770
 772#   772     774#   774     776#   776     778#   778     780#   780     782#   782     784#   784
 786#   786     789#   789     791#   791     793#   793     795#   795     797#   797     799#   799
 801#   801     803#   803     805#   805     807#   807     809#   809     811#   811     813#   813
 815#   815     817#   817     819#   819     822#   822     824#   824     826#   826     828#   828
 830#   830     832#   832     834#   834     836#   836     838#   838     840#   840     842#   842
 844#   844     846#   846     848#   848     850#   850     852#   852     857#   857     859#   859
 862#   862     864#   864     866#   866     868#   868     870#   870     872#   872     874#   874
 876#   876     878#   878     880#   880     882#   882     884#   884     886#   886     888#   888
 890#   890     892#   892     895#   895     897#   897     899#   899     901#   901     903#   903
 905#   905     907#   907     909#   909     911#   911     913#   913     915#   915     917#   917
 919#   919     921#   921     923#   923     925#   925     928#   928     930#   930     932#   932
 934#   934     936#   936     938#   938     940#   940     942#   942     944#   944     946#   946
 948#   948     950#   950     952#   952     954#   954     956#   956     958#   958     961#   961
 963#   963     965#   965     967#   967     969#   969     971#   971     973#   973     975#   975
 977#   977     979#   979     981#   981     983#   983     985#   985     987#   987     989#   989
 991#   991     996#   996     998#   998    1001#  1001    1003#  1003    1005#  1005    1007#  1007
1009#  1009    1011#  1011    1013#  1013    1015#  1015    1017#  1017    1019#  1019    1021#  1021
1023#  1023    1025#  1025    1027#  1027    1029#  1029    1031#  1031    1034#  1034    1036#  1036
1038#  1038    1040#  1040    1042#  1042    1044#  1044    1046#  1046    1048#  1048    1050#  1050
1052#  1052    1054#  1054    1056#  1056    1058#  1058    1060#  1060    1062#  1062    1064#  1064
1067#  1067    1069#  1069    1071#  1071    1073#  1073    1075#  1075    1077#  1077    1079#  1079
1081#  1081    1083#  1083    1085#  1085    1087#  1087    1089#  1089    1091#  1091    1093#  1093
1095#  1095    1097#  1097    1100#  1100    1102#  1102    1104#  1104    1106#  1106    1108#  1108
1110#  1110    1112#  1112    1114#  1114    1116#  1116    1118#  1118    1120#  1120    1122#  1122
1124#  1124    1126#  1126    1128#  1128    1130#  1130    1135#  1135    1137#  1137    1140#  1140
1142#  1142    1144#  1144    1146#  1146    1148#  1148    1150#  1150    1152#  1152    1154#  1154
```

```
1156#  1156   1158#  1158   1160#  1160   1162#  1162   1164#  1164   1166#  1166   1168#  1168    SEQ 1254
1170#  1170   1173#  1173   1175#  1175   1177#  1177   1179#  1179   1181#  1181   1183#  1183
1185#  1185   1187#  1187   1189#  1189   1191#  1191   1193#  1193   1195#  1195   1197#  1197
1199#  1199   1201#  1201   1203#  1203   1206#  1206   1208#  1208   1210#  1210   1212#  1212
1214#  1214   1216#  1216   1218#  1218   1220#  1220   1222#  1222   1224#  1224   1226#  1226
1228#  1228   1230#  1230   1232#  1232   1234#  1234   1236#  1236   1239#  1239   1241#  1241
1243#  1243   1245#  1245   1247#  1247   1249#  1249   1251#  1251   1253#  1253   1255#  1255
1257#  1257   1259#  1259   1261#  1261   1263#  1263   1265#  1265   1267#  1267   1269#  1269
1274#  1274   1276#  1276   1279#  1279   1281#  1281   1283#  1283   1285#  1285   1287#  1287
1289#  1289   1291#  1291   1293#  1293   1295#  1295   1297#  1297   1299#  1299   1301#  1301
1303#  1303   1305#  1305   1307#  1307   1309#  1309   1312#  1312   1314#  1314   1316#  1316
1318#  1318   1320#  1320   1322#  1322   1324#  1324   1326#  1326   1328#  1328   1330#  1330
1332#  1332   1334#  1334   1336#  1336   1338#  1338   1340#  1340   1342#  1342   1345#  1345
1347#  1347   1349#  1349   1351#  1351   1353#  1353   1355#  1355   1357#  1357   1359#  1359
1361#  1361   1363#  1363   1365#  1365   1367#  1367   1369#  1369   1371#  1371   1373#  1373
1375#  1375   1378#  1378   1380#  1380   1382#  1382   1384#  1384   1386#  1386   1388#  1388
1390#  1390   1392#  1392   1394#  1394   1396#  1396   1398#  1398   1400#  1400   1402#  1402
1404#  1404   1406#  1406   1408#  1408   1413#  1413   1415#  1415   1418#  1418   1420#  1420
1422#  1422   1424#  1424   1426#  1426   1428#  1428   1430#  1430   1432#  1432   1434#  1434
1436#  1436   1438#  1438   1440#  1440   1442#  1442   1444#  1444   1446#  1446   1448#  1448
1451#  1451   1453#  1453   1455#  1455   1457#  1457   1459#  1459   1461#  1461   1463#  1463
1465#  1465   1467#  1467   1469#  1469   1471#  1471   1473#  1473   1475#  1475   1477#  1477
1479#  1479   1481#  1481   1484#  1484   1486#  1486   1488#  1488   1490#  1490   1492#  1492
1494#  1494   1496#  1496   1498#  1498   1500#  1500   1502#  1502   1504#  1504   1506#  1506
1508#  1508   1510#  1510   1512#  1512   1514#  1514   1517#  1517   1519#  1519   1521#  1521
1523#  1523   1525#  1525   1527#  1527   1529#  1529   1531#  1531   1533#  1533   1535#  1535
1537#  1537   1539#  1539   1541#  1541   1543#  1543   1545#  1545   1547#  1547   1552#  1552
1554#  1554   1557#  1557   1559#  1559   1561#  1561   1563#  1563   1565#  1565   1567#  1567
1569#  1569   1571#  1571   1573#  1573   1575#  1575   1577#  1577   1579#  1579   1581#  1581
1583#  1583   1585#  1585   1587#  1587   1590#  1590   1592#  1592   1594#  1594   1596#  1596
1598#  1598   1600#  1600   1602#  1602   1604#  1604   1606#  1606   1608#  1608   1610#  1610
1612#  1612   1614#  1614   1616#  1616   1618#  1618   1620#  1620   1623#  1623   1625#  1625
1627#  1627   1629#  1629   1631#  1631   1633#  1633   1635#  1635   1637#  1637   1639#  1639
1641#  1641   1643#  1643   1645#  1645   1647#  1647   1649#  1649   1651#  1651   1653#  1653
1656#  1656   1658#  1658   1660#  1660   1662#  1662   1664#  1664   1666#  1666   1668#  1668
1670#  1670   1672#  1672   1674#  1674   1676#  1676   1678#  1678   1680#  1680   1682#  1682
1684#  1684   1686#  1686   1691#  1691   1693#  1693   1696#  1696   1698#  1698   1700#  1700
1702#  1702   1704#  1704   1706#  1706   1708#  1708   1710#  1710   1712#  1712   1714#  1714
1716#  1716   1718#  1718   1720#  1720   1722#  1722   1724#  1724   1726#  1726   1729#  1729
1731#  1731   1733#  1733   1735#  1735   1737#  1737   1739#  1739   1741#  1741   1743#  1743
1745#  1745   1747#  1747   1749#  1749   1751#  1751   1753#  1753   1755#  1755   1757#  1757
1759#  1759   1762#  1762   1764#  1764   1766#  1766   1768#  1768   1770#  1770   1772#  1772
1774#  1774   1776#  1776   1778#  1778   1780#  1780   1782#  1782   1784#  1784   1786#  1786
1788#  1788   1790#  1790   1792#  1792   1795#  1795   1797#  1797   1799#  1799   1801#  1801
1803#  1803   1805#  1805   1807#  1807   1809#  1809   1811#  1811   1813#  1813   1815#  1815
1817#  1817   1819#  1819   1821#  1821   1823#  1823   1825#  1825   1830#  1830   1832#  1832
1835#  1835   1837#  1837   1839#  1839   1841#  1841   1843#  1843   1845#  1845   1847#  1847
1849#  1849   1851#  1851   1853#  1853   1855#  1855   1857#  1857   1859#  1859   1861#  1861
1863#  1863   1865#  1865   1868#  1868   1870#  1870   1872#  1872   1874#  1874   1876#  1876
1878#  1878   1880#  1880   1882#  1882   1884#  1884   1886#  1886   1888#  1888   1890#  1890
1892#  1892   1894#  1894   1896#  1896   1898#  1898   1901#  1901   1903#  1903   1905#  1905
1907#  1907   1909#  1909   1911#  1911   1913#  1913   1915#  1915   1917#  1917   1919#  1919
1921#  1921   1923#  1923   1925#  1925   1927#  1927   1929#  1929   1931#  1931   1934#  1934
```

```
    1936# 1936  1938# 1938  1940# 1940  1942# 1942  1944# 1944  1946# 1946  1948# 1948    SEQ 1255
    1950# 1950  1952# 1952  1954# 1954  1956# 1956  1958# 1958  1960# 1960  1962# 1962
    1964# 1964  1969# 1969  1971# 1971  1974# 1974  1976# 1976  1978# 1978  1980# 1980
    1982# 1982  1984# 1984  1986# 1986  1988# 1988  1990# 1990  1992# 1992  1994# 1994
    1996# 1996  1998# 1998  2000# 2000  2002# 2002  2004# 2004  2007# 2007  2009# 2009
    2011# 2011  2013# 2013  2015# 2015  2017# 2017  2019# 2019  2021# 2021  2023# 2023
    2025# 2025  2027# 2027  2029# 2029  2031# 2031  2033# 2033  2035# 2035  2037# 2037
    2040# 2040  2042# 2042  2044# 2044  2046# 2046  2048# 2048  2050# 2050  2052# 2052
    2054# 2054  2056# 2056  2058# 2058  2060# 2060  2062# 2062  2064# 2064  2066# 2066
    2068# 2068  2070# 2070  2073# 2073  2075# 2075  2077# 2077  2079# 2079  2081# 2081
    2083# 2083  2085# 2085  2087# 2087  2089# 2089  2091# 2091  2093# 2093  2095# 2095
    2097# 2097  2099# 2099  2101# 2101  2103# 2103  2108# 2108  2110# 2110  2113# 2113
    2115# 2115  2117# 2117  2119# 2119  2121# 2121  2123# 2123  2125# 2125  2127# 2127
    2129# 2129  2131# 2131  2133# 2133  2135# 2135  2137# 2137  2139# 2139  2141# 2141
    2143# 2143  2146# 2146  2148# 2148  2150# 2150  2152# 2152  2154# 2154  2156# 2156
    2158# 2158  2160# 2160  2162# 2162  2164# 2164  2166# 2166  2168# 2168  2170# 2170
    2172# 2172  2174# 2174  2176# 2176  2179# 2179  2181# 2181  2183# 2183  2185# 2185
    2187# 2187  2189# 2189  2191# 2191  2193# 2193  2195# 2195  2197# 2197  2199# 2199
    2201# 2201  2203# 2203  2205# 2205  2207# 2207  2209# 2209  2212# 2212  2214# 2214
    2216# 2216  2218# 2218  2220# 2220  2222# 2222  2224# 2224  2226# 2226  2228# 2228
    2230# 2230  2232# 2232  2234# 2234  2236# 2236  2238# 2238  2240# 2240  2242# 2242
    2247# 2247  2249# 2249  2252# 2252  2254# 2254  2256# 2256  2258# 2258  2260# 2260
    2262# 2262  2264# 2264  2266# 2266  2268# 2268  2270# 2270  2272# 2272  2274# 2274
    2276# 2276  2278# 2278  2280# 2280  2282# 2282  2285# 2285  2287# 2287  2289# 2289
    2291# 2291  2293# 2293  2295# 2295  2297# 2297  2299# 2299  2301# 2301  2303# 2303
    2305# 2305  2307# 2307  2309# 2309  2311# 2311  2313# 2313  2315# 2315  2318# 2318
    2320# 2320  2322# 2322  2324# 2324  2326# 2326  2328# 2328  2330# 2330  2332# 2332
    2334# 2334  2336# 2336  2338# 2338  2340# 2340  2342# 2342  2344# 2344  2346# 2346
    2348# 2348  2351# 2351  2353# 2353  2355# 2355  2357# 2357  2359# 2359  2361# 2361
    2363# 2363  2365# 2365  2367# 2367  2369# 2369  2371# 2371  2373# 2373  2375# 2375
    2377# 2377  2379# 2379  2381# 2381  2386# 2386  2388# 2388  2391# 2391  2393# 2393
    2395# 2395  2397# 2397  2399# 2399  2401# 2401  2403# 2403  2405# 2405  2407# 2407
    2409# 2409  2411# 2411  2413# 2413  2415# 2415  2417# 2417  2419# 2419  2421# 2421
    2424# 2424  2426# 2426  2428# 2428  2430# 2430  2432# 2432  2434# 2434  2436# 2436
    2438# 2438  2440# 2440  2442# 2442  2444# 2444  2446# 2446  2448# 2448  2450# 2450
    2452# 2452  2454# 2454  2457# 2457  2459# 2459  2461# 2461  2463# 2463  2465# 2465
    2467# 2467  2469# 2469  2471# 2471  2473# 2473  2475# 2475  2477# 2477  2479# 2479
    2481# 2481  2483# 2483  2485# 2485  2487# 2487  2490# 2490  2492# 2492  2494# 2494
    2496# 2496  2498# 2498  2500# 2500  2502# 2502  2504# 2504  2506# 2506  2508# 2508
    2510# 2510  2512# 2512  2514# 2514  2516# 2516  2518# 2518  2520# 2520  2522# 2522
.LA  159   168   177   186   195   204   213   222   231   240   249   258   267   276
     285   303   306   308   310   312   314   316   318   320   322   324   326   328
     330   332   334   336   339   341   343   345   347   349   351   353   355   357
     359   361   363   365   367   369   372   374   376   378   380   382   384   386
     388   390   392   394   396   398   400   402   405   407   409   411   413   415
     417   419   421   423   425   427   429   431   433   435   442   445   447   449
     451   453   455   457   459   461   463   465   467   469   471   473   475   478
     480   482   484   486   488   490   492   494   496   498   500   502   504   506
     508   511   513   515   517   519   521   523   525   527   529   531   533   535
     537   539   541   544   546   548   550   552   554   556   558   560   562   564
     566   568   570   572   574   581   584   586   588   590   592   594   596   598
     600   602   604   606   608   610   612   614   617   619   621   623   625   627
     629   631   633   635   637   639   641   643   645   647   650   652   654   656
```

```
 658  660  662  664  666  668  670  672  674  676  678  680  683  685     SEQ 1256
 687  689  691  693  695  697  699  701  703  705  707  709  711  713
 720  723  725  727  729  731  733  735  737  739  741  743  745  747
 749  751  753  756  758  760  762  764  766  768  770  772  774  776
 778  780  782  784  786  789  791  793  795  797  799  801  803  805
 807  809  811  813  815  817  819  822  824  826  828  830  832  834
 836  838  840  842  844  846  848  850  852  859  862  864  866  868
 870  872  874  876  878  880  882  884  886  888  890  892  895  897
 899  901  903  905  907  909  911  913  915  917  919  921  923  925
 928  930  932  934  936  938  940  942  944  946  948  950  952  954
 956  958  961  963  965  967  969  971  973  975  977  979  981  983
 985  987  989  991  998 1001 1003 1005 1007 1009 1011 1013 1015 1017
1019 1021 1023 1025 1027 1029 1031 1034 1036 1038 1040 1042 1044 1046
1048 1050 1052 1054 1056 1058 1060 1062 1064 1067 1069 1071 1073 1075
1077 1079 1081 1083 1085 1087 1089 1091 1093 1095 1097 1100 1102 1104
1106 1108 1110 1112 1114 1116 1118 1120 1122 1124 1126 1128 1130 1137
1140 1142 1144 1146 1148 1150 1152 1154 1156 1158 1160 1162 1164 1166
1168 1170 1173 1175 1177 1179 1181 1183 1185 1187 1189 1191 1193 1195
1197 1199 1201 1203 1206 1208 1210 1212 1214 1216 1218 1220 1222 1224
1226 1228 1230 1232 1234 1236 1239 1241 1243 1245 1247 1249 1251 1253
1255 1257 1259 1261 1263 1265 1267 1269 1276 1279 1281 1283 1285 1287
1289 1291 1293 1295 1297 1299 1301 1303 1305 1307 1309 1312 1314 1316
1318 1320 1322 1324 1326 1328 1330 1332 1334 1336 1338 1340 1342 1345
1347 1349 1351 1353 1355 1357 1359 1361 1363 1365 1367 1369 1371 1373
1375 1378 1380 1382 1384 1386 1388 1390 1392 1394 1396 1398 1400 1402
1404 1406 1408 1415 1418 1420 1422 1424 1426 1428 1430 1432 1434 1436
1438 1440 1442 1444 1446 1448 1451 1453 1455 1457 1459 1461 1463 1465
1467 1469 1471 1473 1475 1477 1479 1481 1484 1486 1488 1490 1492 1494
1496 1498 1500 1502 1504 1506 1508 1510 1512 1514 1517 1519 1521 1523
1525 1527 1529 1531 1533 1535 1537 1539 1541 1543 1545 1547 1554 1557
1559 1561 1563 1565 1567 1569 1571 1573 1575 1577 1579 1581 1583 1585
1587 1590 1592 1594 1596 1598 1600 1602 1604 1606 1608 1610 1612 1614
1616 1618 1620 1623 1625 1627 1629 1631 1633 1635 1637 1639 1641 1643
1645 1647 1649 1651 1653 1656 1658 1660 1662 1664 1666 1668 1670 1672
1674 1676 1678 1680 1682 1684 1686 1693 1696 1698 1700 1702 1704 1706
1708 1710 1712 1714 1716 1718 1720 1722 1724 1726 1729 1731 1733 1735
1737 1739 1741 1743 1745 1747 1749 1751 1753 1755 1757 1759 1762 1764
1766 1768 1770 1772 1774 1776 1778 1780 1782 1784 1786 1788 1790 1792
1795 1797 1799 1801 1803 1805 1807 1809 1811 1813 1815 1817 1819 1821
1823 1825 1832 1835 1837 1839 1841 1843 1845 1847 1849 1851 1853 1855
1857 1859 1861 1863 1865 1868 1870 1872 1874 1876 1878 1880 1882 1884
1886 1888 1890 1892 1894 1896 1898 1901 1903 1905 1907 1909 1911 1913
1915 1917 1919 1921 1923 1925 1927 1929 1931 1934 1936 1938 1940 1942
1944 1946 1948 1950 1952 1954 1956 1958 1960 1962 1964 1971 1974 1976
1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004
2007 2009 2011 2013 2015 2017 2019 2021 2023 2025 2027 2029 2031 2033
2035 2037 2040 2042 2044 2046 2048 2050 2052 2054 2056 2058 2060 2062
2064 2066 2068 2070 2073 2075 2077 2079 2081 2083 2085 2087 2089 2091
2093 2095 2097 2099 2101 2103 2110 2113 2115 2117 2119 2121 2123 2125
2127 2129 2131 2133 2135 2137 2139 2141 2143 2146 2148 2150 2152 2154
2156 2158 2160 2162 2164 2166 2168 2170 2172 2174 2176 2179 2181 2183
2185 2187 2189 2191 2193 2195 2197 2199 2201 2203 2205 2207 2209 2212
2214 2216 2218 2220 2222 2224 2226 2228 2230 2232 2234 2236 2238 2240
```

```
        2242  2249  2252  2254  2256  2258  2260  2262  2264  2266  2268  2270  2272  2274     SEQ 1257
        2276  2278  2280  2282  2285  2287  2289  2291  2293  2295  2297  2299  2301  2303
        2305  2307  2309  2311  2313  2315  2318  2320  2322  2324  2326  2328  2330  2332
        2334  2336  2338  2340  2342  2344  2346  2348  2351  2353  2355  2357  2359  2361
        2363  2365  2367  2369  2371  2373  2375  2377  2379  2381  2388  2391  2393  2395
        2397  2399  2401  2403  2405  2407  2409  2411  2413  2415  2417  2419  2421  2424
        2426  2428  2430  2432  2434  2436  2438  2440  2442  2444  2446  2448  2450  2452
        2454  2457  2459  2461  2463  2465  2467  2469  2471  2473  2475  2477  2479  2481
        2483  2485  2487  2490  2492  2494  2496  2498  2500  2502  2504  2506  2508  2510
        2512  2514  2516  2518  2520
.LADDR   103   150   294   296   301
.LAND    303   322   351   388   417   442   461   490   527   556   581   600   629   666
         695   720   739   768   805   834   859   878   907   944   973   998  1017  1046
        1083  1112  1137  1156  1185  1222  1251  1276  1295  1324  1361  1390  1415  1434
        1463  1500  1529  1554  1573  1602  1639  1668  1693  1712  1741  1778  1807  1832
        1851  1880  1917  1946  1971  1990  2019  2056  2085  2110  2129  2158  2195  2224
        2249  2268  2297  2334  2363  2388  2407  2436  2473  2502
.LB      111   113   115   117   119   121   123   125   127   129   131   133   135   137
         139   141   143   303   306   310   314   318   322   326   330   334   339   343
         347   351   355   359   363   367   372   376   380   384   388   392   396   400
         405   409   413   417   421   425   429   433   442   445   449   453   457   461
         465   469   473   478   482   486   490   494   498   502   506   511   515   519
         523   527   531   535   539   544   548   552   556   560   564   568   572   581
         584   588   592   596   600   604   608   612   617   621   625   629   633   637
         641   645   650   654   658   662   666   670   674   678   683   687   691   695
         699   703   707   711   720   723   727   731   735   739   743   747   751   756
         760   764   768   772   776   780   784   789   793   797   801   805   809   813
         817   822   826   830   834   838   842   846   850   859   862   866   870   874
         878   882   886   890   895   899   903   907   911   915   919   923   928   932
         936   940   944   948   952   956   961   965   969   973   977   981   985   989
         998  1001  1005  1009  1013  1017  1021  1025  1029  1034  1038  1042  1046  1050
        1054  1058  1062  1067  1071  1075  1079  1083  1087  1091  1095  1100  1104  1108
        1112  1116  1120  1124  1128  1137  1140  1144  1148  1152  1156  1160  1164  1168
        1173  1177  1181  1185  1189  1193  1197  1201  1206  1210  1214  1218  1222  1226
        1230  1234  1239  1243  1247  1251  1255  1259  1263  1267  1276  1279  1283  1287
        1291  1295  1299  1303  1307  1312  1316  1320  1324  1328  1332  1336  1340  1345
        1349  1353  1357  1361  1365  1369  1373  1378  1382  1386  1390  1394  1398  1402
        1406  1415  1418  1422  1426  1430  1434  1438  1442  1446  1451  1455  1459  1463
        1467  1471  1475  1479  1484  1488  1492  1496  1500  1504  1508  1512  1517  1521
        1525  1529  1533  1537  1541  1545  1554  1557  1561  1565  1569  1573  1577  1581
        1585  1590  1594  1598  1602  1606  1610  1614  1618  1623  1627  1631  1635  1639
        1643  1647  1651  1656  1660  1664  1668  1672  1676  1680  1684  1693  1696  1700
        1704  1708  1712  1716  1720  1724  1729  1733  1737  1741  1745  1749  1753  1757
        1762  1766  1770  1774  1778  1782  1786  1790  1795  1799  1803  1807  1811  1815
        1819  1823  1832  1835  1839  1843  1847  1851  1855  1859  1863  1868  1872  1876
        1880  1884  1888  1892  1896  1901  1905  1909  1913  1917  1921  1925  1929  1934
        1938  1942  1946  1950  1954  1958  1962  1971  1974  1978  1982  1986  1990  1994
        1998  2002  2007  2011  2015  2019  2023  2027  2031  2035  2040  2044  2048  2052
        2056  2060  2064  2068  2073  2077  2081  2085  2089  2093  2097  2101  2110  2113
        2117  2121  2125  2129  2133  2137  2141  2146  2150  2154  2158  2162  2166  2170
        2174  2179  2183  2187  2191  2195  2199  2203  2207  2212  2216  2220  2224  2228
        2232  2236  2240  2249  2252  2256  2260  2264  2268  2272  2276  2280  2285  2289
        2293  2297  2301  2305  2309  2313  2318  2322  2326  2330  2334  2338  2342  2346
```

```
           2351   2355   2359   2363   2367   2371   2375   2379   2388   2391   2395   2399   2403   2407     SEQ 1258
           2411   2415   2419   2424   2428   2432   2436   2440   2444   2448   2452   2457   2461   2465
           2469   2473   2477   2481   2485   2490   2494   2498   2502   2506   2510   2514   2518
.LBAD       152    161    170    179    188    197    206    215    224    233    242    251    260    269
            278    287    294    296
.LCCFZ      150    159    168    177    186    195    204    213    222    231    240    249    258    267
            276    285
.LCENA      150    159    168    177    186    195    204    213    222    231    240    249    258    267
            276    285
.LCJS       301    308    312    316    320    324    328    332    336    341    345    349    353    357
            361    365    369    374    378    382    386    390    394    398    402    407    411    415
            419    423    427    431    435    440    447    451    455    459    463    467    471    475
            480    484    488    492    496    500    504    508    513    517    521    525    529    533
            537    541    546    550    554    558    562    566    570    574    579    586    590    594
            598    602    606    610    614    619    623    627    631    635    639    643    647    652
            656    660    664    668    672    676    680    685    689    693    697    701    705    709
            713    718    725    729    733    737    741    745    749    753    758    762    766    770
            774    778    782    786    791    795    799    803    807    811    815    819    824    828
            832    836    840    844    848    852    857    864    868    872    876    880    884    888
            892    897    901    905    909    913    917    921    925    930    934    938    942    946
            950    954    958    963    967    971    975    979    983    987    991    996   1003   1007
           1011   1015   1019   1023   1027   1031   1036   1040   1044   1048   1052   1056   1060   1064
           1069   1073   1077   1081   1085   1089   1093   1097   1102   1106   1110   1114   1118   1122
           1126   1130   1135   1142   1146   1150   1154   1158   1162   1166   1170   1175   1179   1183
           1187   1191   1195   1199   1203   1208   1212   1216   1220   1224   1228   1232   1236   1241
           1245   1249   1253   1257   1261   1265   1269   1274   1281   1285   1289   1293   1297   1301
           1305   1309   1314   1318   1322   1326   1330   1334   1338   1342   1347   1351   1355   1359
           1363   1367   1371   1375   1380   1384   1388   1392   1396   1400   1404   1408   1413   1420
           1424   1428   1432   1436   1440   1444   1448   1453   1457   1461   1465   1469   1473   1477
           1481   1486   1490   1494   1498   1502   1506   1510   1514   1519   1523   1527   1531   1535
           1539   1543   1547   1552   1559   1563   1567   1571   1575   1579   1583   1587   1592   1596
           1600   1604   1608   1612   1616   1620   1625   1629   1633   1637   1641   1645   1649   1653
           1658   1662   1666   1670   1674   1678   1682   1686   1691   1698   1702   1706   1710   1714
           1718   1722   1726   1731   1735   1739   1743   1747   1751   1755   1759   1764   1768   1772
           1776   1780   1784   1788   1792   1797   1801   1805   1809   1813   1817   1821   1825   1830
           1837   1841   1845   1849   1853   1857   1861   1865   1870   1874   1878   1882   1886   1890
           1894   1898   1903   1907   1911   1915   1919   1923   1927   1931   1936   1940   1944   1948
           1952   1956   1960   1964   1969   1976   1980   1984   1988   1992   1996   2000   2004   2009
           2013   2017   2021   2025   2029   2033   2037   2042   2046   2050   2054   2058   2062   2066
           2070   2075   2079   2083   2087   2091   2095   2099   2103   2108   2115   2119   2123   2127
           2131   2135   2139   2143   2148   2152   2156   2160   2164   2168   2172   2176   2181   2185
           2189   2193   2197   2201   2205   2209   2214   2218   2222   2226   2230   2234   2238   2242
           2247   2254   2258   2262   2266   2270   2274   2278   2282   2287   2291   2295   2299   2303
           2307   2311   2315   2320   2324   2328   2332   2336   2340   2344   2348   2353   2357   2361
           2365   2369   2373   2377   2381   2386   2393   2397   2401   2405   2409   2413   2417   2421
           2426   2430   2434   2438   2442   2446   2450   2454   2459   2463   2467   2471   2475   2479
           2483   2487   2492   2496   2500   2504   2508   2512   2516   2520
.LCJV       150    159    168    177    186    195    204    213    222    231    240    249    258    267
            276    285
.LCONT      111    113    115    117    119    121    123    125    127    129    131    133    135    137
            139    141    143    303    306    310    314    318    322    326    330    334    339    343
            347    351    355    359    363    367    372    376    380    384    388    392    396    400
            405    409    413    417    421    425    429    433    442    445    449    453    457    461
            465    469    473    478    482    486    490    494    498    502    506    511    515    519
            523    527    531    535    539    544    548    552    556    560    564    568    572    581
            584    588    592    596    600    604    608    612    617    621    625    629    633    637
            641    645    650    654    658    662    666    670    674    678    683    687    691    695
            699    703    707    711    720    723    727    731    735    739    743    747    751    756
            760    764    768    772    776    780    784    789    793    797    801    805    809    813
            817    822    826    830    834    838    842    846    850    859    862    866    870    874
            878    882    886    890    895    899    903    907    911    915    919    923    928    932
            936    940    944    948    952    956    961    965    969    973    977    981    985    989
            998   1001   1005   1009   1013   1017   1021   1025   1029   1034   1038   1042   1046   1050
           1054   1058   1062   1067   1071   1075   1079   1083   1087   1091   1095   1100   1104   1108
           1112   1116   1120   1124   1128   1137   1140   1144   1148   1152   1156   1160   1164   1168
           1173   1177   1181   1185   1189   1193   1197   1201   1206   1210   1214   1218   1222   1226
           1230   1234   1239   1243   1247   1251   1255   1259   1263   1267   1276   1279   1283   1287
           1291   1295   1299   1303   1307   1312   1316   1320   1324   1328   1332   1336   1340   1345
           1349   1353   1357   1361   1365   1369   1373   1378   1382   1386   1390   1394   1398   1402
           1406   1415   1418   1422   1426   1430   1434   1438   1442   1446   1451   1455   1459   1463
           1467   1471   1475   1479   1484   1488   1492   1496   1500   1504   1508   1512   1517   1521
           1525   1529   1533   1537   1541   1545   1554   1557   1561   1565   1569   1573   1577   1581
           1585   1590   1594   1598   1602   1606   1610   1614   1618   1623   1627   1631   1635   1639
           1643   1647   1651   1656   1660   1664   1668   1672   1676   1680   1684   1693   1696   1700
           1704   1708   1712   1716   1720   1724   1729   1733   1737   1741   1745   1749   1753   1757
           1762   1766   1770   1774   1778   1782   1786   1790   1795   1799   1803   1807   1811   1815
           1819   1823   1832   1835   1839   1843   1847   1851   1855   1859   1863   1868   1872   1876
           1880   1884   1888   1892   1896   1901   1905   1909   1913   1917   1921   1925   1929   1934
           1938   1942   1946   1950   1954   1958   1962   1971   1974   1978   1982   1986   1990   1994
           1998   2002   2007   2011   2015   2019   2023   2027   2031   2035   2040   2044   2048   2052
           2056   2060   2064   2068   2073   2077   2081   2085   2089   2093   2097   2101   2110   2113
           2117   2121   2125   2129   2133   2137   2141   2146   2150   2154   2158   2162   2166   2170
           2174   2179   2183   2187   2191   2195   2199   2203   2207   2212   2216   2220   2224   2228
           2232   2236   2240   2249   2252   2256   2260   2264   2268   2272   2276   2280   2285   2289
           2293   2297   2301   2305   2309   2313   2318   2322   2326   2330   2334   2338   2342   2346
           2351   2355   2359   2363   2367   2371   2375   2379   2388   2391   2395   2399   2403   2407
           2411   2415   2419   2424   2428   2432   2436   2440   2444   2448   2452   2457   2461   2465
           2469   2473   2477   2481   2485   2490   2494   2498   2502   2506   2510   2514   2518
.LCRTN      145    154    163    172    181    190    199    208    217    226    235    244    253    262
            271    280    289
.LD         111    113    115    117    119    121    123    125    127    129    131    133    135    137
            139    141    143    145    150    159    168    177    186    195    204    213    222    231
            240    249    258    267    276    285    303    306    308    310    312    314    316    318
            320    322    324    326    328    330    332    334    336    339    341    343    345    347
            349    351    353    355    357    359    361    363    365    367    369    372    374    376
            378    380    382    384    386    388    390    392    394    396    398    400    402    405
            407    409    411    413    415    417    419    421    423    425    427    429    431    433
            435    442    445    447    449    451    453    455    457    459    461    463    465    467
            469    471    473    475    478    480    482    484    486    488    490    492    494    496
            498    500    502    504    506    508    511    513    515    517    519    521    523    525
            527    529    531    533    535    537    539    541    544    546    548    550    552    554
            556    558    560    562    564    566    568    570    572    574    581    584    586    588
            590    592    594    596    598    600    602    604    606    608    610    612    614    617
            619    621    623    625    627    629    631    633    635    637    639    641    643    645
            647    650    652    654    656    658    660    662    664    666    668    670    672    674
            676    678    680    683    685    687    689    691    693    695    697    699    701    703
```

SEQ 1260

705
739
768
797
826
859
888
917
946
975
1009
1038
1067
1095
1124
1158
1187
1216
1245
1279
1307
1336
1365
1394
1428
1457
1486
1514
1543
1577
1606
1635
1664
1698
1726
1755
1784
1813
1847
1876
1905
1934
1962
1996
2025
2054
2083
2117
2146
2174
2203
2232
2266

707
741
770
799
828
862
890
919
948
977
1011
1040
1069
1097
1126
1160
1189
1218
1247
1281
1309
1338
1367
1396
1430
1459
1488
1517
1545
1579
1608
1637
1666
1700
1729
1757
1786
1815
1849
1878
1907
1936
1964
1998
2027
2056
2085
2119
2148
2176
2205
2234
2268

709
743
772
801
830
864
892
921
950
979
1013
1042
1071
1100
1128
1162
1191
1220
1249
1283
1312
1340
1369
1398
1432
1461
1490
1519
1547
1581
1610
1639
1668
1702
1731
1759
1788
1817
1851
1880
1909
1938
1971
2000
2029
2058
2087
2121
2150
2179
2207
2236
2270

711
745
774
803
832
866
895
923
952
981
1015
1044
1073
1102
1130
1164
1193
1222
1251
1285
1314
1342
1371
1400
1434
1463
1492
1521
1554
1583
1612
1641
1670
1704
1733
1762
1790
1819
1853
1882
1911
1940
1974
2002
2031
2060
2089
2123
2152
2181
2209
2238
2272

713
747
776
805
834
868
897
925
954
983
1017
1046
1075
1104
1137
1166
1195
1224
1253
1287
1316
1345
1373
1402
1436
1465
1494
1523
1557
1585
1614
1643
1672
1706
1735
1764
1792
1821
1855
1884
1913
1942
1976
2004
2033
2062
2091
2125
2154
2183
2212
2240
2274

720
749
778
807
836
870
899
928
956
985
1019
1048
1077
1106
1140
1168
1197
1226
1255
1289
1318
1347
1375
1404
1438
1467
1496
1525
1559
1587
1616
1645
1674
1708
1737
1766
1795
1823
1857
1886
1915
1944
1978
2007
2035
2064
2093
2127
2156
2185
2214
2242
2276

723
751
780
809
838
872
901
930
958
987
1021
1050
1079
1108
1142
1170
1199
1228
1257
1291
1320
1349
1378
1406
1440
1469
1498
1527
1561
1590
1618
1647
1676
1710
1739
1768
1797
1825
1859
1888
1917
1946
1980
2009
2037
2066
2095
2129
2158
2187
2216
2249
2278

725
753
782
811
840
874
903
932
961
989
1023
1052
1081
1110
1144
1173
1201
1230
1259
1293
1322
1351
1380
1408
1442
1471
1500
1529
1563
1592
1620
1649
1678
1712
1741
1770
1799
1832
1861
1890
1919
1948
1982
2011
2040
2068
2097
2131
2160
2189
2218
2252
2280

727
756
784
813
842
876
905
934
963
991
1025
1054
1083
1112
1146
1175
1203
1232
1261
1295
1324
1353
1382
1415
1444
1473
1502
1531
1565
1594
1623
1651
1680
1714
1743
1772
1801
1835
1863
1892
1921
1950
1984
2013
2042
2070
2099
2133
2162
2191
2220
2254
2282

729
758
786
815
844
878
907
936
965
998
1027
1056
1085
1114
1148
1177
1206
1234
1263
1297
1326
1355
1384
1418
1446
1475
1504
1533
1567
1596
1625
1653
1682
1716
1745
1774
1803
1837
1865
1894
1923
1952
1986
2015
2044
2073
2101
2135
2164
2193
2222
2256
2285

731
760
789
817
846
880
909
938
967
1001
1029
1058
1087
1116
1150
1179
1208
1236
1265
1299
1328
1357
1386
1420
1448
1477
1506
1535
1569
1598
1627
1656
1684
1718
1747
1776
1805
1839
1868
1896
1925
1954
1988
2017
2046
2075
2103
2137
2166
2195
2224
2258
2287

733
762
791
819
848
882
911
940
969
1003
1031
1060
1089
1118
1152
1181
1210
1239
1267
1301
1330
1359
1388
1422
1451
1479
1508
1537
1571
1600
1629
1658
1686
1720
1749
1778
1807
1841
1870
1898
1927
1956
1990
2019
2048
2077
2110
2139
2168
2197
2226
2260
2289

735
764
793
822
850
884
913
942
971
1005
1034
1062
1091
1120
1154
1183
1212
1241
1269
1303
1332
1361
1390
1424
1453
1481
1510
1539
1573
1602
1631
1660
1693
1722
1751
1780
1809
1843
1872
1901
1929
1958
1992
2021
2050
2079
2113
2141
2170
2199
2228
2262
2291

737
766
795
824
852
886
915
944
973
1007
1036
1064
1093
1122
1156
1185
1214
1243
1276
1305
1334
1363
1392
1426
1455
1484
1512
1541
1575
1604
1633
1662
1696
1724
1753
1782
1811
1845
1874
1903
1931
1960
1994
2023
2052
2081
2115
2143
2172
2201
2230
2264
2293

```
            2295  2297  2299  2301  2303  2305  2307  2309  2311  2313  2315  2318  2320  2322    SEQ 1261
            2324  2326  2328  2330  2332  2334  2336  2338  2340  2342  2344  2346  2348  2351
            2353  2355  2357  2359  2361  2363  2365  2367  2369  2371  2373  2375  2377  2379
            2381  2388  2391  2393  2395  2397  2399  2401  2403  2405  2407  2409  2411  2413
            2415  2417  2419  2421  2424  2426  2428  2430  2432  2434  2436  2438  2440  2442
            2444  2446  2448  2450  2452  2454  2457  2459  2461  2463  2465  2467  2469  2471
            2473  2475  2477  2479  2481  2483  2485  2487  2490  2492  2494  2496  2498  2500
            2502  2504  2506  2508  2510  2512  2514  2516  2518  2520
.LJ          103   150   152   159   161   168   170   177   179   186   188   195   197   204
             206   213   215   222   224   231   233   240   242   249   251   258   260   267
             269   276   278   285   287   294   296   301   308   312   316   320   324   328
             332   336   341   345   349   353   357   361   365   369   374   378   382   386
             390   394   398   402   407   411   415   419   423   427   431   435   440   447
             451   455   459   463   467   471   475   480   484   488   492   496   500   504
             508   513   517   521   525   529   533   537   541   546   550   554   558   562
             566   570   574   579   586   590   594   598   602   606   610   614   619   623
             627   631   635   639   643   647   652   656   660   664   668   672   676   680
             685   689   693   697   701   705   709   713   718   725   729   733   737   741
             745   749   753   758   762   766   770   774   778   782   786   791   795   799
             803   807   811   815   819   824   828   832   836   840   844   848   852   857
             864   868   872   876   880   884   888   892   897   901   905   909   913   917
             921   925   930   934   938   942   946   950   954   958   963   967   971   975
             979   983   987   991   996  1003  1007  1011  1015  1019  1023  1027  1031  1036
            1040  1044  1048  1052  1056  1060  1064  1069  1073  1077  1081  1085  1089  1093
            1097  1102  1106  1110  1114  1118  1122  1126  1130  1135  1142  1146  1150  1154
            1158  1162  1166  1170  1175  1179  1183  1187  1191  1195  1199  1203  1208  1212
            1216  1220  1224  1228  1232  1236  1241  1245  1249  1253  1257  1261  1265  1269
            1274  1281  1285  1289  1293  1297  1301  1305  1309  1314  1318  1322  1326  1330
            1334  1338  1342  1347  1351  1355  1359  1363  1367  1371  1375  1380  1384  1388
            1392  1396  1400  1404  1408  1413  1420  1424  1428  1432  1436  1440  1444  1448
            1453  1457  1461  1465  1469  1473  1477  1481  1486  1490  1494  1498  1502  1506
            1510  1514  1519  1523  1527  1531  1535  1539  1543  1547  1552  1559  1563  1567
            1571  1575  1579  1583  1587  1592  1596  1600  1604  1608  1612  1616  1620  1625
            1629  1633  1637  1641  1645  1649  1653  1658  1662  1666  1670  1674  1678  1682
            1686  1691  1698  1702  1706  1710  1714  1718  1722  1726  1731  1735  1739  1743
            1747  1751  1755  1759  1764  1768  1772  1776  1780  1784  1788  1792  1797  1801
            1805  1809  1813  1817  1821  1825  1830  1837  1841  1845  1849  1853  1857  1861
            1865  1870  1874  1878  1882  1886  1890  1894  1898  1903  1907  1911  1915  1919
            1923  1927  1931  1936  1940  1944  1948  1952  1956  1960  1964  1969  1976  1980
            1984  1988  1992  1996  2000  2004  2009  2013  2017  2021  2025  2029  2033  2037
            2042  2046  2050  2054  2058  2062  2066  2070  2075  2079  2083  2087  2091  2095
            2099  2103  2108  2115  2119  2123  2127  2131  2135  2139  2143  2148  2152  2156
            2160  2164  2168  2172  2176  2181  2185  2189  2193  2197  2201  2205  2209  2214
            2218  2222  2226  2230  2234  2238  2242  2247  2254  2258  2262  2266  2270  2274
            2278  2282  2287  2291  2295  [illegible]  2303  2307  2311  2315  2320  2324  2328  2332
            2336  2340  2344  2348  2353  2357  2361  [illegible]  2369  2373  2377  2381  2386  2393
            2397  2401  2405  2409  2413  2417  2421  [illegible]  2430  2434  2438  2442  2446  2450
            2454  2459  2463  2467  2471  2475  2479  [illegible]  2487  2492  2496  2500  2504  2508
            2512  2516  2520  2522
.LJMAP       103   152   161   170   179   188   197   206   215   224   233   242   251   260
             269   278   287   294   296  2522
.LJZ         105
.LMGC        111   113   115   117   119   121   123   125   127   129   131   133   135   137
```

```
           139   141   143   306   310   314   318   322   326   330   334   339   343   347      SEQ 1262
           351   355   359   363   367   372   376   380   384   388 [illegible] 396   400   405
           409   413   417   421   425   429   433   445   449   453 [illegible] 461   465   469
           473   478   482   486   490   494   498   502   506   511 [illegible] 519   523   527
           531   535   539   544   548   552   556   560   564   568   572   584   588   592
           596   600   604   608   612   617   621   625   629   633   637   641   645   650
           654   658   662   666   670   674   678   683   687   691   695   699   703   707
           711   723   727   731   735   739   743   747   751   756   760   764   768   772
           776   780   784   789   793   797   801   805   809   813   817   822   826   830
           834   838   842   846   850   862   866   870   874   878   882   886   890   895
           899   903   907   911   915   919   923   928   932   936   940   944   948   952
           956   961   965   969   973   977   981   985   989  1001  1005  1009  1013  1017
          1021  1025  1029  1034  1038  1042  1046  1050  1054  1058  1062  1067  1071  1075
          1079  1083  1087  1091  1095  1100  1104  1108  1112  1116  1120  1124  1128  1140
          1144  1148  1152  1156  1160  1164  1168  1173  1177  1181  1185  1189  1193  1197
          1201  1206  1210  1214  1218  1222  1226  1230  1234  1239  1243  1247  1251  1255
          1259  1263  1267  1279  1283  1287  1291  1295  1299  1303  1307  1312  1316  1320
          1324  1328  1332  1336  1340  1345  1349  1353  1357  1361  1365  1369  1373  1378
          1382  1386  1390  1394  1398  1402  1406  1418  1422  1426  1430  1434  1438  1442
          1446  1451  1455  1459  1463  1467  1471  1475  1479  1484  1488  1492  1496  1500
          1504  1508  1512  1517  1521  1525  1529  1533  1537  1541  1545  1557  1561  1565
          1569  1573  1577  1581  1585  1590  1594  1598  1602  1606  1610  1614  1618  1623
          1627  1631  1635  1639  1643  1647  1651  1656  1660  1664  1668  1672  1676  1680
          1684  1696  1700  1704  1708  1712  1716  1720  1724  1729  1733  1737  1741  1745
          1749  1753  1757  1762  1766  1770  1774  1778  1782  1786  1790  1795  1799  1803
          1807  1811  1815  1819  1823  1835  1839  1843  1847  1851  1855  1859  1863  1868
          1872  1876  1880  1884  1888  1892  1896  1901  1905  1909  1913  1917  1921  1925
          1929  1934  1938  1942  1946  1950  1954  1958  1962  1974  1978  1982  1986  1990
          1994  1998  2002  2007  2011  2015  2019  2023  2027  2031  2035  2040  2044  2048
          2052  2056  2060  2064  2068  2073  2077  2081  2085  2089  2093  2097  2101  2113
          2117  2121  2125  2129  2133  2137  2141  2146  2150  2154  2158  2162  2166  2170
          2174  2179  2183  2187  2191  2195  2199  2203  2207  2212  2216  2220  2224  2228
          2232  2236  2240  2252  2256  2260  2264  2268  2272  2276  2280  2285  2289  2293
          2297  2301  2305  2309  2313  2318  2322  2326  2330  2334  2338  2342  2346  2351
          2355  2359  2363  2367  2371  2375  2379  2391  2395  2399  2403  2407  2411  2415
          2419  2424  2428  2432  2436  2440  2444  2448  2452  2457  2461  2465  2469  2473
          2477  2481  2485  2490  2494  2498  2502  2506  2510  2514  2518
.LNAND     326   347   392   413   465   486   531   552   604   625   670   691   743   764
           809   830   882   903   948   969  1021  1042  1087  1108  1160  1181  1226  1247
          1299  1320  1365  1386  1438  1459  1504  1525  1577  1598  1643  1664  1716  1737
          1782  1803  1855  1876  1921  1942  1994  2015  2060  2081  2133  2154  2199  2220
          2272  2293  2338  2359  2411  2432  2477  2498
.LOR       111   113   115   117   119   121   123   125   127   129   131   133   135   137
           139   141   143   150   159   168   177   186   195   204   213   222   231   240
           249   258   267   276   285   308   312   316   318   320   324   328   332   336
           341   345   349   353   355   357   361   365   369   374   378   382   384   386
           390   394   398   402   407   411   415   419   421   423   427   431   435   447
           451   455   457   459   463   467   471   475   480   484   488   492   494   496
           500   504   508   513   517   521   523   525   529   533   537   541   546   550
           554   558   560   562   566   570   574   586   590   594   596   598   602   606
           610   614   619   623   627   631   633   635   639   643   647   652   656   660
           662   664   668   672   676   680   685   689   693   697   699   701   705   709
           713   725   729   733   735   737   741   745   749   753   758   762   766   770
```

```
             772   774   778   782   786   791   795   799   801   803   807   811   815   819
             824   828   832   836   838   840   844   848   852   864   868   872   874   876
             880   884   888   892   897   901   905   909   911   913   917   921   925   930
             934   938   940   942   946   950   954   958   963   967   971   975   977   979
             983   987   991  1003  1007  1011  1013  1015  1019  1023  1027  1031  1036  1040
            1044  1048  1050  1052  1056  1060  1064  1069  1073  1077  1079  1081  1085  1089
            1093  1097  1102  1106  1110  1114  1116  1118  1122  1126  1130  1142  1146  1150
            1152  1154  1158  1162  1166  1170  1175  1179  1183  1187  1189  1191  1195  1199
            1203  1208  1212  1216  1218  1220  1224  1228  1232  1236  1241  1245  1249  1253
            1255  1257  1261  1265  1269  1281  1285  1289  1291  1293  1297  1301  1305  1309
            1314  1318  1322  1326  1328  1330  1334  1338  1342  1347  1351  1355  1357  1359
            1363  1367  1371  1375  1380  1384  1388  1392  1394  1396  1400  1404  1408  1420
            1424  1428  1430  1432  1436  1440  1444  1448  1453  1457  1461  1465  1467  1469
            1473  1477  1481  1486  1490  1494  1496  1498  1502  1506  1510  1514  1519  1523
            1527  1531  1533  1535  1539  1543  1547  1559  1563  1567  1569  1571  1575  1579
            1583  1587  1592  1596  1600  1604  1606  1608  1612  1616  1620  1625  1629  1633
            1635  1637  1641  1645  1649  1653  1658  1662  1666  1670  1672  1674  1678  1682
            1686  1698  1702  1706  1708  1710  1714  1718  1722  1726  1731  1735  1739  1743
            1745  1747  1751  1755  1759  1764  1768  1772  1774  1776  1780  1784  1788  1792
            1797  1801  1805  1809  1811  1813  1817  1821  1825  1837  1841  1845  1847  1849
            1853  1857  1861  1865  1870  1874  1878  1882  1884  1886  1890  1894  1898  1903
            1907  1911  1913  1915  1919  1923  1927  1931  1936  1940  1944  1948  1950  1952
            1956  1960  1964  1976  1980  1984  1986  1988  1992  1996  2000  2004  2009  2013
            2017  2021  2023  2025  2029  2033  2037  2042  2046  2050  2052  2054  2058  2062
            2066  2070  2075  2079  2083  2087  2089  2091  2095  2099  2103  2115  2119  2123
            2125  2127  2131  2135  2139  2143  2148  2152  2156  2160  2162  2164  2168  2172
            2176  2181  2185  2189  2191  2193  2197  2201  2205  2209  2214  2218  2222  2226
            2228  2230  2234  2238  2242  2254  2258  2262  2264  2266  2270  2274  2278  2282
            2287  2291  2295  2299  2301  2303  2307  2311  2315  2320  2324  2328  2330  2332
            2336  2340  2344  2348  2353  2357  2361  2365  2367  2369  2373  2377  2381  2393
            2397  2401  2403  2405  2409  2413  2417  2421  2426  2430  2434  2438  2440  2442
            2446  2450  2454  2459  2463  2467  2469  2471  2475  2479  2483  2487  2492  2496
            2500  2504  2506  2508  2512  2516  2520
.LPLUS       306   367   372   433   445   506   511   572   584   645   650   711   723   784
             789   850   862   923   928   989  1001  1062  1067  1128  1140  1201  1206  1267
            1279  1340  1345  1406  1418  1479  1484  1545  1557  1618  1623  1684  1696  1757
            1762  1823  1835  1896  1901  1962  1974  2035  2040  2101  2113  2174  2179  2240
            2252  2313  2318  2379  2391  2452  2457  2518
.LRMIN       314   359   380   425   453   498   519   564   592   637   658   703   731   776
             797   842   870   915   936   981  1009  1054  1075  1120  1148  1193  1214  1259
            1287  1332  1353  1398  1426  1471  1492  1537  1565  1610  1631  1676  1704  1749
            1770  1815  1843  1888  1909  1954  1982  2027  2048  2093  2121  2166  2187  2232
            2260  2305  2326  2371  2399  2444  2465  2510
.LSOA        150   159   168   177   186   195   204   213   222   231   240   249   258   267
             276   285   303   308   312   316   320   324   328   332   336   341   345   349
             353   357   361   365   369   372   374   378   380   382   386   388   390   394
             396   398   402   407   411   415   419   423   427   431   435   442   447   451
             455   459   463   467   471   475   480   484   488   492   496   500   504   508
             511   513   517   519   521   525   527   529   533   535   537   541   546   550
             554   558   562   566   570   574   581   586   590   594   598   602   606   610
             614   619   623   627   631   635   639   643   647   650   652   655   658   660
             664   666   668   672   674   676   680   685   689   693   697   701   705   709
             713   720   725   729   733   737   741   745   749   753   758   762   766   770
            774     778     782     786     789     791     795     797     799     803     805     807     811     813     SEQ 1264
            815     819     824     828     832     836     840     844     848     852     859     864     868     872
            876     880     884     888     892     897     901     905     909     913     917     921     925     928
            930     934     936     938     942     944     946     950     952     954     958     963     967     971
            975     979     983     987     991     998    1003    1007    1011    1015    1019    1023    1027    1031
           1036    1040    1044    1048    1052    1056    1060    1064    1067    1069    1073    1075    1077    1081
           1083    1085    1089    1091    1093    1097    1102    1106    1110    1114    1118    1122    1126    1130
           1137    1142    1146    1150    1154    1158    1162    1166    1170    1175    1179    1183    1187    1191
           1195    1199    1203    1206    1208    1212    1214    1216    1220    1222    1224    1228    1230    1232
           1236    1241    1245    1249    1253    1257    1261    1265    1269    1276    1281    1285    1289    1293
           1297    1301    1305    1309    1314    1318    1322    1326    1330    1334    1338    1342    1345    1347
           1351    1353    1355    1359    1361    1363    1367    1369    1371    1375    1380    1384    1388    1392
           1396    1400    1404    1408    1415    1420    1424    1428    1432    1436    1440    1444    1448    1453
           1457    1461    1465    1469    1473    1477    1481    1484    1486    1490    1492    1494    1498    1500
           1502    1506    1508    1510    1514    1519    1523    1527    1531    1535    1539    1543    1547    1554
           1559    1563    1567    1571    1575    1579    1583    1587    1592    1596    1600    1604    1608    1612
           1616    1620    1623    1625    1629    1631    1633    1637    1639    1641    1645    1647    1649    1653
           1658    1662    1666    1670    1674    1678    1682    1686    1693    1698    1702    1706    1710    1714
           1718    1722    1726    1731    1735    1739    1743    1747    1751    1755    1759    1762    1764    1768
           1770    1772    1776    1778    1780    1784    1786    1788    1792    1797    1801    1805    1809    1813
           1817    1821    1825    1832    1837    1841    1845    1849    1853    1857    1861    1865    1870    1874
           1878    1882    1886    1890    1894    1898    1901    1903    1907    1909    1911    1915    1917    1919
           1923    1925    1927    1931    1936    1940    1944    1948    1952    1956    1960    1964    1971    1976
           1980    1984    1988    1992    1996    2000    2004    2009    2013    2017    2021    2025    2029    2033
           2037    2040    2042    2046    2048    2050    2054    2056    2058    2062    2064    2066    2070    2075
           2079    2083    2087    2091    2095    2099    2103    2110    2115    2119    2123    2127    2131    2135
           2139    2143    2148    2152    2156    2160    2164    2168    2172    2176    2179    2181    2185    2187
           2189    2193    2195    2197    2201    2203    2205    2209    2214    2218    2222    2226    2230    2234
           2238    2242    2249    2254    2258    2262    2266    2270    2274    2278    2282    2287    2291    2295
           2299    2303    2307    2311    2315    2318    2320    2324    2326    2328    2332    2334    2336    2340
           2342    2344    2348    2353    2357    2361    2365    2369    2373    2377    2381    2388    2393    2397
           2401    2405    2409    2413    2417    2421    2426    2430    2434    2438    2442    2446    2450    2454
           2457    2459    2463    2465    2467    2471    2473    2475    2479    2481    2483    2487    2492    2496
           2500    2504    2508    2512    2516    2520
.LSOB       343     351     359     367     482     490     498     506     621     629     637     645     760     768
            776     784     899     907     915     923    1038    1046    1054    1062    1177    1185    1193    1201
           1316    1324    1332    1340    1455    1463    1471    1479    1594    1602    1610    1618    1733    1741
           1749    1757    1872    1880    1888    1896    2011    2019    2027    2035    2150    2158    2166    2174
           2289    2297    2305    2313    2428    2436    2444    2452
.LSOQ       339     347     355     363     478     486     494     502     617     625     633     641     756     764
            772     780     895     903     911     919    1034    1042    1050    1058    1173    1181    1189    1197
           1312    1320    1328    1336    1451    1459    1467    1475    1590    1598    1606    1614    1729    1737
           1745    1753    1868    1876    1884    1892    2007    2015    2023    2031    2146    2154    2162    2170
           2285    2293    2301    2309    2424    2432    2440    2448
.LSAB       310     318     326     334     449     457     465     473     588     596     604     612     727     735
            743     751     866     874     882     890    1005    1013    1021    1029    1144    1152    1160    1168
           1283    1291    1299    1307    1422    1430    1438    1446    1561    1569    1577    1585    1700    1708
           1716    1724    1839    1847    1855    1863    1978    1986    1994    2002    2117    2125    2133    2141
           2256    2264    2272    2280    2395    2403    2411    2419
.LSAQ       306     314     322     330     445     453     461     469     584     592     600     608     723     731
            739     747     862     870     878     886    1001    1009    1017    1025    1140    1148    1156    1164
           1279    1287    1295    1303    1418    1426    1434    1442    1557    1565    1573    1581    1696    1704
           1712    1720    1835    1843    1851    1859    1974    1982    1990    1998    2113    2121    2129    2137
```

```
          2252  2260  2268  2276  2391  2399  2407  2415
.LSDO      111   113   115   117   119   121   123   125   127   129   131   133   135   137
           139   141   143   409   417   425   433   548   556   564   572   687   695   703
           711   826   834   842   850   965   973   981   989  1104  1112  1120  1128  1243
          1251  1259  1267  1382  1390  1398  1406  1521  1529  1537  1545  1660  1668  1676
          1684  1799  1807  1815  1823  1938  1946  1954  1962  2077  2085  2093  2101  2216
          2224  2232  2240  2355  2363  2371  2379  2494  2502  2510  2518
.LSDA      376   384   392   400   515   523   531   539   654   662   670   678   793   801
           809   817   932   940   948   956  1071  1079  1087  1095  1210  1218  1226  1234
          1349  1357  1365  1373  1488  1496  1504  1512  1627  1635  1643  1651  1766  1774
          1782  1790  1905  1913  1921  1929  2044  2052  2060  2068  2183  2191  2199  2207
          2322  2330  2338  2346  2461  2469  2477  2485
.LSDQ      405   413   421   429   544   552   560   568   683   691   699   707   822   830
           838   846   961   969   977   985  1100  1108  1116  1124  1239  1247  1255  1263
          1378  1386  1394  1402  1517  1525  1533  1541  1656  1664  1672  1680  1795  1803
          1811  1819  1934  1942  1950  1958  2073  2081  2089  2097  2212  2220  2228  2236
          2351  2359  2367  2375  2490  2498  2506  2514
.LSKCN     111   113   115   117   119   121   123   125   127   129   131   133   135   137
           139   141   143   306   310   314   318   322   326   330   334   339   343   347
           351   355   359   363   367   372   376   380   384   388   392   396   400   405
           409   413   417   421   425   429   433   445   449   453   457   461   465   469
           473   478   482   486   490   494   498   502   506   511   515   519   523   527
           531   535   539   544   548   552   556   560   564   568   572   584   588   592
           596   600   604   608   612   617   621   625   629   633   637   641   645   650
           654   658   662   666   670   674   678   683   687   691   695   699   703   707
           711   723   727   731   735   739   743   747   751   756   760   764   768   772
           776   780   784   789   793   797   801   805   809   813   817   822   826   830
           834   838   842   846   850   862   866   870   874   878   882   886   890   895
           899   903   907   911   915   919   923   928   932   936   940   944   948   952
           956   961   965   969   973   977   981   985   989  1001  1005  1009  1013  1017
          1021  1025  1029  1034  1038  1042  1046  1050  1054  1058  1062  1067  1071  1075
          1079  1083  1087  1091  1095  1100  1104  1108  1112  1116  1120  1124  1128  1140
          1144  1148  1152  1156  1160  1164  1168  1173  1177  1181  1185  1189  1193  1197
          1201  1206  1210  1214  1218  1222  1226  1230  1234  1239  1243  1247  1251  1255
          1259  1263  1267  1279  1283  1287  1291  1295  1299  1303  1307  1312  1316  1320
          1324  1328  1332  1336  1340  1345  1349  1353  1357  1361  1365  1369  1373  1378
          1382  1386  1390  1394  1398  1402  1406  1418  1422  1426  1430  1434  1438  1442
          1446  1451  1455  1459  1463  1467  1471  1475  1479  1484  1488  1492  1496  1500
          1504  1508  1512  1517  1521  1525  1529  1533  1537  1541  1545  1557  1561  1565
          1569  1573  1577  1581  1585  1590  1594  1598  1602  1606  1610  1614  1618  1623
          1627  1631  1635  1639  1643  1647  1651  1656  1660  1664  1668  1672  1676  1680
          1684  1696  1700  1704  1708  1712  1716  1720  1724  1729  1733  1737  1741  1745
          1749  1753  1757  1762  1766  1770  1774  1778  1782  1786  1790  1795  1799  1803
          1807  1811  1815  1819  1823  1835  1839  1843  1847  1851  1855  1859  1863  1868
          1872  1876  1880  1884  1888  1892  1896  1901  1905  1909  1913  1917  1921  1925
          1929  1934  1938  1942  1946  1950  1954  1958  1962  1974  1978  1982  1986  1990
          1994  1998  2002  2007  2011  2015  2019  2023  2027  2031  2035  2040  2044  2048
          2052  2056  2060  2064  2068  2073  2077  2081  2085  2089  2093  2097  2101  2113
          2117  2121  2125  2129  2133  2137  2141  2146  2150  2154  2158  2162  2166  2170
          2174  2179  2183  2187  2191  2195  2199  2203  2207  2212  2216  2220  2224  2228
          2232  2236  2240  2252  2256  2260  2264  2268  2272  2276  2280  2285  2289  2293
          2297  2301  2305  2309  2313  2318  2322  2326  2330  2334  2338  2342  2346  2351
          2355  2359  2363  2367  2371  2375  2379  2391  2395  2399  2403  2407  2411  2415
```

```
         2419 2424 2428 2432 2436 2440 2444 2448 2452 2457 2461 2465 2469 2473      SEQ 1266
         2477 2481 2485 2490 2494 2498 2502 2506 2510 2514 2518
.LSMIN    310  363  376  429  449  502  515  568  588  641  654  707  727  780
          793  846  866  919  932  985 1005 1058 1071 1124 1144 1197 1210 1263
         1283 1336 1349 1402 1422 1475 1488 1541 1561 1614 1627 1680 1700 1753
         1766 1819 1839 1892 1905 1958 1978 2031 2044 2097 2117 2170 2183 2236
         2256 2309 2322 2375 2395 2448 2461 2514
.LXNOR    334  339  400  405  473  478  539  544  612  617  678  683  751  756
          817  822  890  895  956  961 1029 1034 1095 1100 1168 1173 1234 1239
         1307 1312 1373 1378 1446 1451 1512 1517 1585 1590 1651 1656 1724 1729
         1790 1795 1863 1868 1929 1934 2002 2007 2068 2073 2141 2146 2207 2212
         2280 2285 2346 2351 2419 2424 2485 2490
.LXOR     330  343  396  409  469  482  535  548  608  621  674  687  747  760
          813  826  886  899  952  965 1025 1038 1091 1104 1164 1177 1230 1243
         1303 1316 1369 1382 1442 1455 1508 1521 1581 1594 1647 1660 1720 1733
         1786 1799 1859 1872 1925 1938 1998 2011 2064 2077 2137 2150 2203 2216
         2276 2289 2342 2355 2415 2428 2481 2494
.MA       159  168  177  186  195  204  213  222  231  240  249  258  267  276
          285  303  306  308  310  312  314  316  318  320  322  324  326  328
          330  332  334  336  339  341  343  345  347  349  351  353  355  357
          359  361  363  365  367  369  372  374  376  378  380  382  384  386
          388  390  392  394  396  398  400  402  405  407  409  411  413  415
          417  419  421  423  425  427  429  431  433  435  442  445  447  449
          451  453  455  457  459  461  463  465  467  469  471  473  475  478
          480  482  484  486  488  490  492  494  496  498  500  502  504  506
          508  511  513  515  517  519  521  523  525  527  529  531  533  535
          537  539  541  544  546  548  550  552  554  556  558  560  562  564
          566  568  570  572  574  581  584  586  588  590  592  594  596  598
          600  602  604  606  608  610  612  614  617  619  621  623  625  627
          629  631  633  635  637  639  641  643  645  647  650  652  654  656
          658  660  662  664  666  668  670  672  674  676  678  680  683  685
          687  689  691  693  695  697  699  701  703  705  707  709  711  713
          720  723  725  727  729  731  733  735  737  739  741  743  745  747
          749  751  753  756  758  760  762  764  766  768  770  772  774  776
          778  780  782  784  786  789  791  793  795  797  799  801  803  805
          807  809  811  813  815  817  819  822  824  826  828  830  832  834
          836  838  840  842  844  846  848  850  852  859  862  864  866  868
          870  872  874  876  878  880  882  884  886  888  890  892  895  897
          899  901  903  905  907  909  911  913  915  917  919  921  923  925
          928  930  932  934  936  938  940  942  944  946  948  950  952  954
          956  958  961  963  965  967  969  971  973  975  977  979  981  983
          985  987  989  991  998 1001 1003 1005 1007 1009 1011 1013 1015 1017
         1019 1021 1023 1025 1027 1029 1031 1034 1036 1038 1040 1042 1044 1046
         1048 1050 1052 1054 1056 1058 1060 1062 1064 1067 1069 1071 1073 1075
         1077 1079 1081 1083 1085 1087 1089 1091 1093 1095 1097 1100 1102 1104
         1106 1108 1110 1112 1114 1116 1118 1120 1122 1124 1126 1128 1130 1137
         1140 1142 1144 1146 1148 1150 1152 1154 1156 1158 1160 1162 1164 1166
         1168 1170 1173 1175 1177 1179 1181 1183 1185 1187 1189 1191 1193 1195
         1197 1199 1201 1203 1206 1208 1210 1212 1214 1216 1218 1220 1222 1224
         1226 1228 1230 1232 1234 1236 1239 1241 1243 1245 1247 1249 1251 1253
         1255 1257 1259 1261 1263 1265 1267 1269 1276 1279 1281 1283 1285 1287
         1289 1291 1293 1295 1297 1299 1301 1303 1305 1307 1309 1312 1314 1316
         1318 1320 1322 1324 1326 1328 1330 1332 1334 1336 1338 1340 1342 1345
```

```
          1347  1349  1351  1353  1355  1357  1359  1361  1363  1365  1367  1369  1371  1373    SEQ 1267
          1375  1378  1380  1382  1384  1386  1388  1390  1392  1394  1396  1398  1400  1402
          1404  1406  1408  1415  1418  1420  1422  1424  1426  1428  1430  1432  1434  1436
          1438  1440  1442  1444  1446  1448  1451  1453  1455  1457  1459  1461  1463  1465
          1467  1469  1471  1473  1475  1477  1479  1481  1484  1486  1488  1490  1492  1494
          1496  1498  1500  1502  1504  1506  1508  1510  1512  1514  1517  1519  1521  1523
          1525  1527  1529  1531  1533  1535  1537  1539  1541  1543  1545  1547  1554  1557
          1559  1561  1563  1565  1567  1569  1571  1573  1575  1577  1579  1581  1583  1585
          1587  1590  1592  1594  1596  1598  1600  1602  1604  1606  1608  1610  1612  1614
          1616  1618  1620  1623  1625  1627  1629  1631  1633  1635  1637  1639  1641  1643
          1645  1647  1649  1651  1653  1656  1658  1660  1662  1664  1666  1668  1670  1672
          1674  1676  1678  1680  1682  1684  1686  1693  1696  1698  1700  1702  1704  1706
          1708  1710  1712  1714  1716  1718  1720  1722  1724  1726  1729  1731  1733  1735
          1737  1739  1741  1743  1745  1747  1749  1751  1753  1755  1757  1759  1762  1764
          1766  1768  1770  1772  1774  1776  1778  1780  1782  1784  1786  1788  1790  1792
          1795  1797  1799  1801  1803  1805  1807  1809  1811  1813  1815  1817  1819  1821
          1823  1825  1832  1835  1837  1839  1841  1843  1845  1847  1849  1851  1853  1855
          1857  1859  1861  1863  1865  1868  1870  1872  1874  1876  1878  1880  1882  1884
          1886  1888  1890  1892  1894  1896  1898  1901  1903  1905  1907  1909  1911  1913
          1915  1917  1919  1921  1923  1925  1927  1929  1931  1934  1936  1938  1940  1942
          1944  1946  1948  1950  1952  1954  1956  1958  1960  1962  1964  1971  1974  1976
          1978  1980  1982  1984  1986  1988  1990  1992  1994  1996  1998  2000  2002  2004
          2007  2009  2011  2013  2015  2017  2019  2021  2023  2025  2027  2029  2031  2033
          2035  2037  2040  2042  2044  2046  2048  2050  2052  2054  2056  2058  2060  2062
          2064  2066  2068  2070  2073  2075  2077  2079  2081  2083  2085  2087  2089  2091
          2093  2095  2097  2099  2101  2103  2110  2113  2115  2117  2119  2121  2123  2125
          2127  2129  2131  2133  2135  2137  2139  2141  2143  2146  2148  2150  2152  2154
          2156  2158  2160  2162  2164  2166  2168  2170  2172  2174  2176  2179  2181  2183
          2185  2187  2189  2191  2193  2195  2197  2199  2201  2203  2205  2207  2209  2212
          2214  2216  2218  2220  2222  2224  2226  2228  2230  2232  2234  2236  2238  2240
          2242  2249  2252  2254  2256  2258  2260  2262  2264  2266  2268  2270  2272  2274
          2276  2278  2280  2282  2285  2287  2289  2291  2293  2295  2297  2299  2301  2303
          2305  2307  2309  2311  2313  2315  2318  2320  2322  2324  2326  2328  2330  2332
          2334  2336  2338  2340  2342  2344  2346  2348  2351  2353  2355  2357  2359  2361
          2363  2365  2367  2369  2371  2373  2375  2377  2379  2381  2388  2391  2393  2395
          2397  2399  2401  2403  2405  2407  2409  2411  2413  2415  2417  2419  2421  2424
          2426  2428  2430  2432  2434  2436  2438  2440  2442  2444  2446  2448  2450  2452
          2454  2457  2459  2461  2463  2465  2467  2469  2471  2473  2475  2477  2479  2481
          2483  2485  2487  2490  2492  2494  2496  2498  2500  2502  2504  2506  2508  2510
          2512  2514  2516  2518  2520
.MAND      303   322   351   388   417   442   461   490   527   556   581   600   629   666
           695   720   739   768   805   834   859   878   907   944   973   998  1017  1046
          1083  1112  1137  1156  1185  1222  1251  1276  1295  1324  1361  1390  1415  1434
          1463  1500  1529  1554  1573  1602  1639  1668  1693  1712  1741  1778  1807  1832
          1851  1880  1917  1946  1971  1990  2019  2056  2085  2110  2129  2158  2195  2224
          2249  2268  2297  2334  2363  2388  2407  2436  2473  2502
.MB        111   113   115   117   119   121   123   125   127   129   131   133   135   137
           139   141   143   303   306   310   314   318   322   326   330   334   339   343
           347   351   355   359   363   367   372   376   380   384   388   392   396   400
           405   409   413   417   421   425   429   433   442   445   449   453   457   461
           465   469   473   478   482   486   490   494   498   502   506   511   515   519
           523   527   531   535   539   544   548   552   556   560   564   568   572   581
           584   588   592   596   600   604   608   612   617   621   625   629   633   637
```

```
            641    645    650    654    658    662    666    670    674    678    683    687    691    695
            699    703    707    711    720    723    727    731    735    739    743    747    751    756
            760    764    768    772    776    780    784    789    793    797    801    805    809    813
            817    822    826    830    834    838    842    846    850    859    862    866    870    874
            878    882    886    890    895    899    903    907    911    915    919    923    928    932
            936    940    944    948    952    956    961    965    969    973    977    981    985    989
            998   1001   1005   1009   1013   1017   1021   1025   1029   1034   1038   1042   1046   1050
           1054   1058   1062   1067   1071   1075   1079   1083   1087   1091   1095   1100   1104   1108
           1112   1116   1120   1124   1128   1137   1140   1144   1148   1152   1156   1160   1164   1168
           1173   1177   1181   1185   1189   1193   1197   1201   1206   1210   1214   1218   1222   1226
           1230   1234   1239   1243   1247   1251   1255   1259   1263   1267   1276   1279   1283   1287
           1291   1295   1299   1303   1307   1312   1316   1320   1324 [illegible] 1332 1336   1340   1345
           1349   1353   1357   1361   1365   1369   1373   1378   1382   1386   1390   1394   1398   1402
           1406   1415   1418   1422   1426   1430   1434   1438   1442   1446   1451   1455   1459   1463
           1467   1471   1475   1479   1484   1488   1492   1496   1500   1504   1508   1512   1517   1521
           1525   1529   1533   1537   1541   1545   1554   1557   1561   1565   1569   1573   1577   1581
           1585   1590   1594   1598   1602   1606   1610   1614   1618   1623   1627   1631   1635   1639
           1643   1647   1651   1656   1660   1664   1668   1672   1676   1680   1684   1693   1696   1700
           1704   1708   1712   1716   1720   1724   1729   1733   1737   1741   1745   1749   1753   1757
           1762   1766   1770   1774   1778   1782   1786   1790   1795   1799   1803   1807   1811   1815
           1819   1823   1832   1835   1839   1843   1847   1851   1855   1859   1863   1868   1872   1876
           1880   1884   1888   1892   1896   1901   1905   1909   1913   1917   1921   1925   1929   1934
           1938   1942   1946   1950   1954   1958   1962   1971   1974   1978   1982   1986   1990   1994
           1998   2002   2007   2011   2015   2019   2023   2027   2031   2035   2040   2044   2048   2052
           2056   2060   2064   2068   2073   2077   2081   2085   2089   2093   2097   2101   2110   2113
           2117   2121   2125   2129   2133   2137   2141   2146   2150   2154   2158   2162   2166   2170
           2174   2179   2183   2187   2191   2195   2199   2203   2207   2212   2216   2220   2224   2228
           2232   2236   2240   2249   2252   2256   2260   2264   2268   2272   2276   2280   2285   2289
           2293   2297   2301   2305   2309   2313   2318   2322   2326   2330   2334   2338   2342   2346
           2351   2355   2359   2363   2367   2371   2375   2379   2388   2391   2395   2399   2403   2407
           2411   2415   2419   2424   2428   2432   2436   2440   2444   2448   2452   2457   2461   2465
           2469   2473   2477   2481   2485   2490   2494   2498   2502   2506   2510   2514   2518
.MBAD       152    161    170    179    188    197    206    215    224    233    242    251    260    269
            278    287    294    296
.MCCFZ      150    159    168    177    186    195    204    213    222    231    240    249    258    267
            276    285
.MCENA      150    159    168    177    186    195    204    213    222    231    240    249    258    267
            276    285
.MCJS       301    308    312    316    320    324    328    332    336    341    345    349    353    357
            361    365    369    374    378    382    386    390    394    398    402    407    411    415
            419    423    427    431    435    440    447    451    455    459    463    467    471    475
            480    484    488    492    496    500    504    508    513    517    521    525    529    533
            537    541    546    550    554    558    562    566    570    574    579    586    590    594
            598    602    606    610    614    619    623    627    631    635    639    643    647    652
            656    660    664    668    672    676    680    685    689    693    697    701    705    709
            713    718    725    729    733    737    741    745    749    753    758    762    766    770
            774    778    782    786    791    795    799    803    807    811    815    819    824    828
            832    836    840    844    848    852    857    864    868    872    876    880    884    888
            892    897    901    905    909    913    917    921    925    930    934    938    942    946
            950    954    958    963    967    971    975    979    983    987    991    996   1003   1007
           1011   1015   1019   1023   1027   1031   1036   1040   1044   1048   1052   1056   1060   1064
           1069   1073   1077   1081   1085   1089   1093   1097   1102   1106   1110   1114   1118   1122
           1126   1130   1135   1142   1146   1150   1154   1158   1162   1166   1170   1175   1179   1183
```

1256

```
          1187  1191  1195  1199  1203  1208  1212  1216  1220  1224  1228  1232  1236  1241      SEQ 1269
          1245  1249  1253  1257  1261  1265  1269  1274  1281  1285  1289  1293  1297  1301
          1305  1309  1314  1318  1322  1326  1330  1334  1338  1342  1347  1351  1355  1359
          1363  1367  1371  1375  1380  1384  1388  1392  1396  1400  1404  1408  1413  1420
          1424  1428  1432  1436  1440  1444  1448  1453  1457  1461  1465  1469  1473  1477
          1481  1486  1490  1494  1498  1502  1506  1510  1514  1519  1523  1527  1531  1535
          1539  1543  1547  1552  1559  1563  1567  1571  1575  1579  1583  1587  1592  1596
          1600  1604  1608  1612  1616  1620  1625  1629  1633  1637  1641  1645  1649  1653
          1658  1662  1666  1670  1674  1678  1682  1686  1691  1698  1702  1706  1710  1714
          1718  1722  1726  1731  1735  1739  1743  1747  1751  1755  1759  1764  1768  1772
          1776  1780  1784  1788  1792  1797  1801  1805  1809  1813  1817  1821  1825  1830
          1837  1841  1845  1849  1853  1857  1861  1865  1870  1874  1878  1882  1886  1890
          1894  1898  1903  1907  1911  1915  1919  1923  1927  1931  1936  1940  1944  1948
          1952  1956  1960  1964  1969  1976  1980  1984  1988  1992  1996  2000  2004  2009
          2013  2017  2021  2025  2029  2033  2037  2042  2046  2050  2054  2058  2062  2066
          2070  2075  2079  2083  2087  2091  2095  2099  2103  2108  2115  2119  2123  2127
          2131  2135  2139  2143  2148  2152  2156  2160  2164  2168  2172  2176  2181  2185
          2189  2193  2197  2201  2205  2209  2214  2218  2222  2226  2230  2234  2238  2242
          2247  2254  2258  2262  2266  2270  2274  2278  2282  2287  2291  2295  2299  2303
          2307  2311  2315  2320  2324  2328  2332  2336  2340  2344  2348  2353  2357  2361
          2365  2369  2373  2377  2381  2386  2393  2397  2401  2405  2409  2413  2417  2421
          2426  2430  2434  2438  2442  2446  2450  2454  2459  2463  2467  2471  2475  2479
          2483  2487  2492  2496  2500  2504  2508  2512  2516  2520
.MCJV      150   159   168   177   186   195   204   213   222   231   240   249   258   267
           276   285
.MCONT     111   113   115   117   119   121   123   125   127   129   131   133   135   137
           139   141   143   303   306   310   314   318   322   326   330   334   339   343
           347   351   355   359   363   367   372   376   380   384   388   392   396   400
           405   409   413   417   421   425   429   433   442   445   449   453   457   461
           465   469   473   478   482   486   490   494   498   502   506   511   515   519
           523   527   531   535   539   544   548   552   556   560   564   568   572   581
           584   588   592   596   600   604   608   612   617   621   625   629   633   637
           641   645   650   654   658   662   666   670   674   678   683   687   691   695
           699   703   707   711   720   723   727   731   735   739   743   747   751   756
           760   764   768   772   776   780   784   789   793   797   801   805   809   813
           817   822   826   830   834   838   842   846   850   859   862   866   870   874
           878   882   886   890   895   899   903   907   911   915   919   923   928   932
           936   940   944   948   952   956   961   965   969   973   977   981   985   989
           998  1001  1005  1009  1013  1017  1021  1025  1029  1034  1038  1042  1046  1050
          1054  1058  1062  1067  1071  1075  1079  1083  1087  1091  1095  1100  1104  1108
          1112  1116  1120  1124  1128  1137  1140  1144  1148  1152  1156  1160  1164  1168
          1173  1177  1181  1185  1189  1193  1197  1201  1206  1210  1214  1218  1222  1226
          1230  1234  1239  1243  1247  1251  1255  1259  1263  1267  1276  1279  1283  1287
          1291  1295  1299  1303  1307  1312  1316  1320  1324  1328  1332  1336  1340  1345
          1349  1353  1357  1361  1365  1369  1373  1378  1382  1386  1390  1394  1398  1402
          1406  1415  1418  1422  1426  1430  1434  1438  1442  1446  1451  1455  1459  1463
          1467  1471  1475  1479  1484  1488  1492  1496  1500  1504  1508  1512  1517  1521
          1525  1529  1533  1537  1541  1545  1554  1557  1561  1565  1569  1573  1577  1581
          1585  1590  1594  1598  1602  1606  1610  1614  1618  1623  1627  1631  1635  1639
          1643  1647  1651  1656  1660  1664  1668  1672  1676  1680  1684  1693  1696  1700
          1704  1708  1712  1716  1720  1724  1729  1733  1737  1741  1745  1749  1753  1757
          1762  1766  1770  1774  1778  1782  1786  1790  1795  1799  1803  1807  1811  1815
          1819  1823  1832  1835  1839  1843  1847  1851  1855  1859  1863  1868  1872  1876
```

.RJ

```
          1880  1884  1888  1892  1896  1901  1905  1909  1913  1917  1921  1925  1929  1934      SEQ 1270
          1938  1942  1946  1950  1954  1958  1962  1971  1974  1978  1982  1986  1990  1994
          1998  2002  2007  2011  2015  2019  2023  2027  2031  2035  2040  2044  2048  2052
          2056  2060  2064  2068  2073  2077  2081  2085  2089  2093  2097  2101  2110  2113
          2117  2121  2125  2129  2133  2137  2141  2146  2150  2154  2158  2162  2166  2170
          2174  2179  2183  2187  2191  2195  2199  2203  2207  2212  2216  2220  2224  2228
          2232  2236  2240  2249  2252  2256  2260  2264  2268  2272  2276  2280  2285  2289
          2293  2297  2301  2305  2309  2313  2318  2322  2326  2330  2334  2338  2342  2346
          2351  2355  2359  2363  2367  2371  2375  2379  2388  2391  2395  2399  2403  2407
          2411  2415  2419  2424  2428  2432  2436  2440  2444  2448  2452  2457  2461  2465
          2469  2473  2477  2481  2485  2490  2494  2498  2502  2506  2510  2514  2518
.MCRTN     145   154   163   172   181   190   199   208   217   226   235   244   253   262
           271   280   289
.MD        111   113   115   117   119   121   123   125   127   129   131   133   135   137
           139   141   143   145   150   159   168   177   186   195   204   213   222   231
           240   249   258   267   276   285   303   306   308   310   312   314   316   318
           320   322   324   326   328   330   332   334   336   339   341   343   345   347
           349   351   353   355   357   359   361   363   365   367   369   372   374   376
           378   380   382   384   386   388   390   392   394   396   398   400   402   405
           407   409   411   413   415   417   419   421   423   425   427   429   431   433
           435   442   445   447   449   451   453   455   457   459   461   463   465   467
           469   471   473   475   478   480   482   484   486   488   490   492   494   496
           498   500   502   504   506   508   511   513   515   517   519   521   523   525
           527   529   531   533   535   537   539   541   544   546   548   550   552   554
           556   558   560   562   564   566   568   570   572   574   581   584   586   588
           590   592   594   596   598   600   602   604   606   608   610   612   614   617
           619   621   623   625   627   629   631   633   635   637   639   641   643   645
           647   650   652   654   656   658   660   662   664   666   668   670   672   674
           676   678   680   683   685   687   689   691   693   695   697   699   701   703
           705   707   709   711   713   720   723   725   727   729   731   733   735   737
           739   741   743   745   747   749   751   753   756   758   760   762   764   766
           768   770   772   774   776   778   780   782   784   786   789   791   793   795
           797   799   801   803   805   807   809   811   813   815   817   819   822   824
           826   828   830   832   834   836   838   840   842   844   846   848   850   852
           859   862   864   866   868   870   872   874   876   878   880   882   884   886
           888   890   892   895   897   899   901   903   905   907   909   911   913   915
           917   919   921   923   925   928   930   932   934   936   938   940   942   944
           946   948   950   952   954   956   958   961   963   965   967   969   971   973
           975   977   979   981   983   985   987   989   991   998  1001  1003  1005  1007
          1009  1011  1013  1015  1017  1019  1021  1023  1025  1027  1029  1031  1034  1036
          1038  1040  1042  1044  1046  1048  1050  1052  1054  1056  1058  1060  1062  1064
          1067  1069  1071  1073  1075  1077  1079  1081  1083  1085  1087  1089  1091  1093
          1095  1097  1100  1102  1104  1106  1108  1110  1112  1114  1116  1118  1120  1122
          1124  1126  1128  1130  1137  1140  1142  1144  1146  1148  1150  1152  1154  1156
          1158  1160  1162  1164  1166  1168  1170  1173  1175  1177  1179  1181  1183  1185
          1187  1189  1191  1193  1195  1197  1199  1201  1203  1206  1208  1210  1212  1214
          1216  1218  1220  1222  1224  1226  1228  1230  1232  1234  1236  1239  1241  1243
          1245  1247  1249  1251  1253  1255  1257  1259  1261  1263  1265  1267  1269  1276
          1279  1281  1283  1285  1287  1289  1291  1293  1295  1297  1299  1301  1303  1305
          1307  1309  1312  1314  1316  1318  1320  1322  1324  1326  1328  1330  1332  1334
          1336  1338  1340  1342  1345  1347  1349  1351  1353  1355  1357  1359  1361  1363
          1365  1367  1369  1371  1373  1375  1378  1380  1382  1384  1386  1388  1390  1392
          1394  1396  1398  1400  1402  1404  1406  1408  1415  1418  1420  1422  1424  1426
```

```
      1428  1430  1432  1434  1436  1438  1440  1442  1444  1446  1448  1451  1453  1455    SEQ 1271
      1457  1459  1461  1463  1465  1467  1469  1471  1473  1475  1477  1479  1481  1484
      1486  1488  1490  1492  1494  1496  1498  1500  1502  1504  1506  1508  1510  1512
      1514  1517  1519  1521  1523  1525  1527  1529  1531  1533  1535  1537  1539  1541
      1543  1545  1547  1554  1557  1559  1561  1563  1565  1567  1569  1571  1573  1575
      1577  1579  1581  1583  1585  1587  1590  1592  1594  1596  1598  1600  1602  1604
      1606  1608  1610  1612  1614  1616  1618  1620  1623  1625  1627  1629  1631  1633
      1635  1637  1639  1641  1643  1645  1647  1649  1651  1653  1656  1658  1660  1662
      1664  1666  1668  1670  1672  1674  1676  1678  1680  1682  1684  1686  1693  1696
      1698  1700  1702  1704  1706  1708  1710  1712  1714  1716  1718  1720  1722  1724
      1726  1729  1731  1733  1735  1737  1739  1741  1743  1745  1747  1749  1751  1753
      1755  1757  1759  1762  1764  1766  1768  1770  1772  1774  1776  1778  1780  1782
      1784  1786  1788  1790  1792  1795  1797  1799  1801  1803  1805  1807  1809  1811
      1813  1815  1817  1819  1821  1823  1825  1832  1835  1837  1839  1841  1843  1845
      1847  1849  1851  1853  1855  1857  1859  1861  1863  1865  1868  1870  1872  1874
      1876  1878  1880  1882  1884  1886  1888  1890  1892  1894  1896  1898  1901  1903
      1905  1907  1909  1911  1913  1915  1917  1919  1921  1923  1925  1927  1929  1931
      1934  1936  1938  1940  1942  1944  1946  1948  1950  1952  1954  1956  1958  1960
      1962  1964  1971  1974  1976  1978  1980  1982  1984  1986  1988  1990  1992  1994
      1996  1998  2000  2002  2004  2007  2009  2011  2013  2015  2017  2019  2021  2023
      2025  2027  2029  2031  2033  2035  2037  2040  2042  2044  2046  2048  2050  2052
      2054  2056  2058  2060  2062  2064  2066  2068  2070  2073  2075  2077  2079  2081
      2083  2085  2087  2089  2091  2093  2095  2097  2099  2101  2103  2110  2113  2115
      2117  2119  2121  2123  2125  2127  2129  2131  2133  2135  2137  2139  2141  2143
      2146  2148  2150  2152  2154  2156  2158  2160  2162  2164  2166  2168  2170  2172
      2174  2176  2179  2181  2183  2185  2187  2189  2191  2193  2195  2197  2199  2201
[illegible]  2205  2207  2209  2212  2214  2216  2218  2220  2222  2224  2226  2228  2230
[illegible]  2234  2236  2238  2240  2242  2249  2252  2254  2256  2258  2260  2262  2264
      2266  2268  2270  2272  2274  2276  2278  2280  2282  2285  2287  2289  2291  2293
      2295  2297  2299  2301  2303  2305  2307  2309  2311  2313  2315  2318  2320  2322
      2324  2326  2328  2330  2332  2334  2336  2338  2340  2342  2344  2346  2348  2351
      2353  2355  2357  2359  2361  2363  2365  2367  2369  2371  2373  2375  2377  2379
      2381  2388  2391  2393  2395  2397  2399  2401  2403  2405  2407  2409  2411  2413
      2415  2417  2419  2421  2424  2426  2428  2430  2432  2434  2436  2438  2440  2442
      2444  2446  2448  2450  2452  2454  2457  2459  2461  2463  2465  2467  2469  2471
      2473  2475  2477  2479  2481  2483  2485  2487  2490  2492  2494  2496  2498  2500
      2502  2504  2506  2508  2510  2512  2514  2516  2518  2520
.MJ    103   150   152   159   161   168   170   177   179   186   188   195   197   204
       206   213   215   222   224   231   233   240   242   249   251   258   260   267
       269   276   278   285   287   294   296   301   308   312   316   320   324   328
       332   336   341   345   349   353   357   361   365   369   374   378   382   386
       390   394   398   402   407   411   415   419   423   427   431   435   440   447
       451   455   459   463   467   471   475   480   484   488   492   496   500   504
       508   513   517   521   525   529   533   537   541   546   550   554   558   562
       566   570   574   579   586   590   594   598   602   606   610   614   619   623
       627   631   635   639   643   647   652   656   660   664   668   672   676   680
       685   689   693   697   701   705   709   713   718   725   729   733   737   741
       745   749   753   758   762   766   770   774   778   782   786   791   795   799
       803   807   811   815   819   824   828   832   836   840   844   848   852   857
       864   868   872   876   880   884   888   892   897   901   905   909   913   917
       921   925   930   934   938   942   946   950   954   958   963   967   971   975
       979   983   987   991   996  1003  1007  1011  1015  1019  1023  1027  1031  1036
      1040  1044  1048  1052  1056  1060  1064  1069  1073  1077  1081  1085  1089  1093
```

```
            1097   1102   1106   1110   1114   1118   1122   1126   1130   1135   1142   1146   1150   1154     SEQ 1272
            1158   1162   1166   1170   1175   1179   1183   1187   1191   1195   1199   1203   1208   1212
            1216   1220   1224   1228   1232   1236   1241   1245   1249   1253   1257   1261   1265   1269
            1274   1281   1285   1289   1293   1297   1301   1305   1309   1314   1318   1322   1326   1330
            1334   1338   1342   1347   1351   1355   1359   1363   1367   1371   1375   1380   1384   1388
            1392   1396   1400   1404   1408   1413   1420   1424   1428   1432   1436   1440   1444   1448
            1453   1457   1461   1465   1469   1473   1477   1481   1486   1490   1494   1498   1502   1506
            1510   1514   1519   1523   1527   1531   1535   1539   1543   1547   1552   1559   1563   1567
            1571   1575   1579   1583   1587   1592   1596   1600   1604   1608   1612   1616   1620   1625
            1629   1633   1637   1641   1645   1649   1653   1658   1662   1666   1670   1674   1678   1682
            1686   1691   1698   1702   1706   1710   1714   1718   1722   1726   1731   1735   1739   1743
            1747   1751   1755   1759   1764   1768   1772   1776   1780   1784   1788   1792   1797   1801
            1805   1809   1813   1817   1821   1825   1830   1837   1841   1845   1849   1853   1857   1861
            1865   1870   1874   1878   1882   1886   1890   1894   1898   1903   1907   1911   1915   1919
            1923   1927   1931   1936   1940   1944   1948   1952   1956   1960   1964   1969   1976   1980
            1984   1988   1992   1996   2000   2004   2009   2013   2017   2021   2025   2029   2033   2037
            2042   2046   2050   2054   2058   2062   2066   2070   2075   2079   2083   2087   2091   2095
            2099   2103   2108   2115   2119   2123   2127   2131   2135   2139   2143   2148   2152   2156
            2160   2164   2168   2172   2176   2181   2185   2189   2193   2197   2201   2205   2209   2214
            2218   2222   2226   2230   2234   2238   2242   2247   2254   2258   2262   2266   2270   2274
            2278   2282   2287   2291   2295   2299   2303   2307   2311   2315   2320   2324   2328   2332
            2336   2340   2344   2348   2353   2357   2361   2365   2369   2373   2377   2381   2386   2393
            2397   2401   2405   2409   2413   2417   2421   2426   2430   2434   2438   2442   2446   2450
            2454   2459   2463   2467   2471   2475   2479   2483   2487   2492   2496   2500   2504   2508
            2512   2516   2520   2522
.MJMAP       103    152    161    170    179    188    197    206    215    224    233    242    251    260
             269    278    287    294    296   2522
.MJZ         105
.MMGC        111    113    115    117    119    121    123    125    127    129    131    133    135    137
             139    141    143    306    310    314    318    322    326    330    334    339    343    347
             351    355    359    363    367    372    376    380    384    388    392    396    400    405
             409    413    417    421    425    429    433    445    449    453    457    461    465    469
             473    478    482    486    490    494    498    502    506    511    515    519    523    527
             531    535    539    544    548    552    556    560    564    568    572    584    588    592
             596    600    604    608    612    617    621    625    629    633    637    641    645    650
             654    658    662    666    670    674    678    683    687    691    695    699    703    707
             711    723    727    731    735    739    743    747    751    756    760    764    768    772
             776    780    784    789    793    797    801    805    809    813    817    822    826    830
             834    838    842    846    850    862    866    870    874    878    882    886    890    895
             899    903    907    911    915    919    923    928    932    936    940    944    948    952
             956    961    965    969    973    977    981    985    989   1001   1005   1009   1013   1017
            1021   1025   1029   1034   1038   1042   1046   1050   1054   1058   1062   1067   1071   1075
            1079   1083   1087   1091   1095   1100   1104   1108   1112   1116   1120   1124   1128   1140
            1144   1148   1152   1156   1160   1164   1168   1173   1177   1181   1185   1189   1193   1197
            1201   1206   1210   1214   1218   1222   1226   1230   1234   1239   1243   1247   1251   1255
            1259   1263   1267   1279   1283   1287   1291   1295   1299   1303   1307   1312   1316   1320
            1324   1328   1332   1336   1340   1345   1349   1353   1357   1361   1365   1369   1373   1378
            1382   1386   1390   1394   1398   1402   1406   1418   1422   1426   1430   1434   1438   1442
            1446   1451   1455   1459   1463   1467   1471   1475   1479   1484   1488   1492   1496   1500
            1504   1508   1512   1517   1521   1525   1529   1533   1537   1541   1545   1557   1561   1565
            1569   1573   1577   1581   1585   1590   1594   1598   1602   1606   1610   1614   1618   1623
            1627   1631   1635   1639   1643   1647   1651   1656   1660   1664   1668   1672   1676   1680
            1684   1696   1700   1704   1708   1712   1716   1720   1724   1729   1733   1737   1741   1745
```

```
           1749   1753   1757   1762   1766   1770   1774   1778   1782   1786   1790   1795   1799   1803   SEQ 1273
           1807   1811   1815   1819   1823   1835   1839   1843   1847   1851   1855   1859   1863   1868
           1872   1876   1880   1884   1888   1892   1896   1901   1905   1909   1913   1917   1921   1925
           1929   1934   1938   1942   1946   1950   1954   1958   1962   1974   1978   1982   1986   1990
           1994   1998   2002   2007   2011   2015   2019   2023   2027   2031   2035   2040   2044   2048
           2052   2056   2060   2064   2068   2073   2077   2081   2085   2089   2093   2097   2101   2113
           2117   2121   2125   2129   2133   2137   2141   2146   2150   2154   2158   2162   2166   2170
           2174   2179   2183   2187   2191   2195   2199   2203   2207   2212   2216   2220   2224   2228
           2232   2236   2240   2252   2256   2260   2264   2268   2272   2276   2280   2285   2289   2293
           2297   2301   2305   2309   2313   2318   2322   2326   2330   2334   2338   2342   2346   2351
           2355   2359   2363   2367   2371   2375   2379   2391   2395   2399   2403   2407   2411   2415
           2419   2424   2428   2432   2436   2440   2444   2448   2452   2457   2461   2465   2469   2473
           2477   2481   2485   2490   2494   2498   2502   2506   2510   2514   2518
.MNAND      326    347    392    413    465    486    531    552    604    625    670    691    743    764
            809    830    882    903    948    969   1021   1042   1087   1108   1160   1181   1226   1247
           1299   1320   1365   1386   1438   1459   1504   1525   1577   1598   1643   1664   1716   1737
           1782   1803   1855   1876   1921   1942   1994   2015   2060   2081   2133   2154   2199   2220
           2272   2293   2338   2359   2411   2432   2477   2498
.MOR        111    113    115    117    119    121    123    125    127    129    131    133    135    137
            139    141    143    150    159    168    177    186    195    204    213    222    231    240
            249    258    267    276    285    308    312    316    318    320    324    328    332    336
            341    345    349    353    355    357    361    365    369    374    378    382    384    386
            390    394    398    402    407    411    415    419    421    423    427    431    435    447
            451    455    457    459    463    467    471    475    480    484    488    492    494    496
            500    504    508    513    517    521    523    525    529    533    537    541    546    550
            554    558    560    562    566    570    574    586    590    594    596    598    602    606
            610    614    619    623    627    631    633    635    639    643    647    652    656    660
            662    664    668    672    676    680    685    689    693    697    699    701    705    709
            713    725    729    733    735    737    741    745    749    753    758    762    766    770
            772    774    778    782    786    791    795    799    801    803    807    811    815    819
            824    828    832    836    838    840    844    848    852    864    868    872    874    876
            880    884    888    892    897    901    905    909    911    913    917    921    925    930
            934    938    940    942    946    950    954    958    963    967    971    975    977    979
            983    987    991   1003   1007   1011   1013   1015   1019   1023   1027   1031   1036   1040
           1044   1048   1050   1052   1056   1060   1064   1069   1073   1077   1079   1081   1085   1089
           1093   1097   1102   1106   1110   1114   1116   1118   1122   1126   1130   1142   1146   1150
           1152   1154   1158   1162   1166   1170   1175   1179   1183   1187   1189   1191   1195   1199
           1203   1208   1212   1216   1218   1220   1224   1228   1232   1236   1241   1245   1249   1253
           1255   1257   1261   1265   1269   1281   1285   1289   1291   1293   1297   1301   1305   1309
           1314   1318   1322   1326   1328   1330   1334   1338   1342   1347   1351   1355   1357   1359
           1363   1367   1371   1375   1380   1384   1388   1392   1394   1396   1400   1404   1408   1420
           1424   1428   1430   1432   1436   1440   1444   1448   1453   1457   1461   1465   1467   1469
           1473   1477   1481   1486   1490   1494   1496   1498   1502   1506   1510   1514   1519   1523
           1527   1531   1533   1535   1539   1543   1547   1559   1563   1567   1569   1571   1575   1579
           1583   1587   1592   1596   1600   1604   1606   1608   1612   1616   1620   1625   1629   1633
           1635   1637   1641   1645   1649   1653   1658   1662   1666   1670   1672   1674   1678   1682
           1686   1698   1702   1706   1708   1710   1714   1718   1722   1726   1731   1735   1739   1743
           1745   1747   1751   1755   1759   1764   1768   1772   1774   1776   1780   1784   1788   1792
           1797   1801   1805   1809   1811   1813   1817   1821   1825   1837   1841   1845   1847   1849
           1853   1857   1861   1865   1870   1874   1878   1882   1884   1886   1890   1894   1898   1903
           1907   1911   1913   1915   1919   1923   1927   1931   1936   1940   1944   1948   1950   [illegible]
           1956   1960   1964   1976   1980   1984   1986   1988   1992   1996   2000   2004   2009   2015
           2017   2021   2023   2025   2029   2033   2037   2042   2046   2050   2052   2054   2058   2062
```

```
            2066   2070   2075   2079   2083   2087   2089   2091   2095   2099   2103   2115   2119   2123     SEQ 1274
            2125   2127   2131   2135   2139   2143   2148   2152   2156   2160   2162   2164   2168   2172
            2176   2181   2185   2189   2191   2193   2197   2201   2205   2209   2214   2218   2222   2226
            2228   2230   2234   2238   2242   2254   2258   2262   2264   2266   2270   2274   2278   2282
            2287   2291   2295   2299   2301   2303   2307   2311   2315   2320   2324   2328   2330   2332
            2336   2340   2344   2348   2353   2357   2361   2365   2367   2369   2373   2377   2381   2393
            2397   2401   2403   2405   2409   2413   2417   2421   2426   2430   2434   2438   2440   2442
     244[illegible]   2450   2454   2459   2463   2467   2469   2471   2475   2479   2483   2487   2492   2496
            2500   2504   2506   2508   2512   2516   2520
.MPLUS       306    367    372    433    445    506    511    572    584    645    650    711    723    784
             789    850    862    923    928    989   1001   1062   1067   1128   1140   1201   1206   1267
            1279   1340   1345   1406   1418   1479   1484   1545   1557   1618   1623   1684   1696   1757
            1762   1823   1835   1896   1901   1962   1974   2035   2040   2101   2113   2174   2179   2240
            2252   2313   2318   2379   2391   2452   2457   2518
.MRMIN       314    359    380    425    453    498    519    564    592    637    658    703    731    776
             797    842    870    915    936    981   1009   1054   1075   1120   1148   1193   1214   1259
            1287   1332   1353   1398   1426   1471   1492   1537   1565   1610   1631   1676   1704   1749
            1770   1815   1843   1888   1909   1954   1982   2027   2048   2093   2121   2166   2187   2232
            2260   2305   2326   2371   2399   2444   2465   2510
.MSOA        150    159    168    177    186    195    204    213    222    231    240    249    258    267
             276    285    303    308    312    316    320    324    328    332    336    341    345    349
             353    357    361    365    369    372    374    378    380    382    386    388    390    394
             396    398    402    407    411    415    419    423    427    431    435    442    447    451
             455    459    463    467    471    475    480    484    488    492    496    500    504    508
             511    513    517    515    521    525    527    529    533    535    537    541    546    550
             554    558    562    566    570    574    581    586    590    594    598    602    606    610
             614    619    623    627    631    635    639    643    647    650    652    656    658    660
             664    666    668    672    674    676    680    685    689    693    697    701    705    709
             713    720    725    729 73[illegible]    737    741    745    749    753    758    762    766    770
             774    778    782    786 [illegible]    791    795    797    799    803    805    807    811    813
             815    819    824    828 [illegible]    836    840    844    848    852    859    864    868    872
             876    880    884    888    892    897    901    905    909    913    917    921    925    928
             930    934    936    938    942    944    946    950    952    954    958    963    967    971
             975    979    983    987    991    998   1003   1007   1011   1015   1019   1023   1027   1031
            1036   1040   1044   1048   1052   1056   1060   1064   1067   1069   1073   1075   1077   1081
            1083   1085   1089   1091   1093   1097   1102   1106   1110   1114   1118   1122   1126   1130
            1137   1142   1146   1150   1154   1158   1162   1166   1170   1175   1179   1183   1187   1191
            1195   1199   1203   1206   1208   1212   1214   1216   1220   1222   1224   1228   1230   1232
            1236   1241   1245   1249   1253   1257   1261   1265   1269   1276   1281   1285   1289   1293
            1297   1301   1305   1309   1314   1318   1322   1326   1330   1334   1338   1342   1345   1347
            1351   1353   1355   1359   1361   1363   1367   1369   1371   1375   1380   1384   1388   1392
            1396   1400   1404   1408   1415   1420   1424   1428   1432   1436   1440   1444   1448   1453
            1457   1461   1465   1469   1473   1477   1481   1484   1486   1490   1492   1494   1498   1500
            1502   1506   1508   1510   1514   1519   1523   1527   1531   1535   1539   1543   1547   1554
            1559   1563   1567   1571   1575   1579   1583   1587   1592   1596   1600   1604   1608   1612
            1616   1620   1623   1625   1629   1631   1633   1637   1639   1641   1645   1647   1649   1653
            1658   1662   1666   1670   1674   1678   1682   1686   1693   1698   1702   1706   1710   1714
            1718   1722   1726   1731   1735   1739   1743   1747   1751   1755   1759   1762   1764   1768
            1770   1772   1776   1778   1780   1784   1786   1788   1792   1797   1801   1805   1809   1813
            1817   1821   1825   1832   1837   1841   1845   1849   1853   1857   1861   1865   1870   1874
            1878   1882   1886   1890   1894   1898   1901   1903   1907   1909   1911   1915   1917   1919
            1923   1925   1927   1931   1936   1940   1944   1948   1952   1956   1960   1964   1971   1976
            1980   1984   1988   1992   1996   2000   2004   2009   2013   2017   2021   2025   2029   2033
```

```
262
            2037   2040   2042   2046   2048   2050   2054   2056   2058   2062   2064   2066   2070   2075     SEQ 1275   CALC
            2079   2083   2087   2091   2095   2099   2103   2110   2115   2119   2123   2127   2131   2135
            2139   2143   2148   2152   2156   2160   2164   2168   2172   2176   2179   2181   2185   2187
            2189   2193   2195   2197   2201   2203   2205   2209   2214   2218   2222   2226   2230   2234
            2238   2242   2249   2254   2258   2262   2266   2270   2274   2278   2282   2287   2291   2295
            2299   2303   2307   2311   2315   2318   2320   2324   2326   2328   2332   2334   2336   2340
            2342   2344   2348   2353   2357   2361   2365   2369   2373   2377   2381   2388   2393   2397
            2401   2405   2409   2413   2417   2421   2426   2430   2434   2438   2442   2446   2450   2454
            2457   2459   2463   2465   2467   2471   2473   2475   2479   2481   2483   2487   2492   2496
            2500   2504   2508   2512   2516   2520
.MSOB        343    351    359    367    482    490    498    506    621    629    637    645    760    768
             776    784    899    907    915    923   1038   1046   1054   1062   1177   1185   1193   1201
            1316   1324   1332   1340   1455   1463   1471   1479   1594   1602   1610   1618   1733   1741
            1749   1757   1872   1880   1888   1896   2011   2019   2027   2035   2150   2158   2166   2174
            2289   2297   2305   2313   2428   2436   2444   2452
.MSOQ        339    347    355    363    478    486    494    502    617    625    633    641    756    764
             772    780    895    903    911    919   1034   1042   1050   1058   1173   1181   1189   1197
            1312   1320   1328   1336   1451   1459   1467   1475   1590   1598   1606   1614   1729   1737
            1745   1753   1868   1876   1884   1892   2007   2015   2023   2031   2146   2154   2162   2170
            2285   2293   2301   2309   2424   2432   2440   2448
.MSAB        310    318    326    334    449    457    465    473    588    596    604    612    727    735
             743    751    866    874    882    890   1005   1013   1021   1029   1144   1152   1160   1168
            1283   1291   1299   1307   1422   1430   1438   1446   1561   1569   1577   1585   1700   1708
            1716   1724   1839   1847   1855   1863   1978   1986   1994   2002   2117   2125   2133   2141
            2256   2264   2272   2280   2395   2403   2411   2419
.MSAQ        306    314    322    330    445    453    461    469    584    592    600    608    723    731
             739    747    862    870    878    886   1001   1009   1017   1025   1140   1148   1156   1164
            1279   1287   1295   1303   1418   1426   1434   1442   1557   1565   1573   1581   1696   1704
            1712   1720   1835   1843   1851   1859   1974   1982   1990   1998   2113   2121   2129   2137
            2252   2260   2268   2276   2391   2399   2407   2415
.MSDO        111    113    115    117    119    121    123    125    127    129    131    133    135    137
             139    141    143    409    417    425    433    548    556    564    572    687    695    703
             711    826    834    842    850    965    973    981    989   1104   1112   1120   1128   1243
            1251   1259   1267   1382   1390   1398   1406   1521   1529   1537   1545   1660   1668   1676
            1684   1799   1807   1815   1823   1938   1946   1954   1962   2077   2085   2093   2101   2216
            2224   2232   2240   2355   2363   2371   2379   2494   2502   2510   2518
.MSDA        376    384    392    400    515    523    531    539    654    662    670    678    793    801
             809    817    932    940    948    956   1071   1079   1087   1095   1210   1218   1226   1234
            1349   1357   1365   1373   1488   1496   1504   1512   1627   1635   1643   1651   1766   1774
            1782   1790   1905   1913   1921   1929   2044   2052   2060   2068   2183   2191   2199   2207
            2322   2330   2338   2346   2461   2469   2477   2485
.MSDQ        405    413    421    429    544    552    560    568    683    691    699    707    822    830
             838    846    961    969    977    985   1100   1108   1116   1124   1239   1247   1255   1263
            1378   1386   1394   1402   1517   1525   1533   1541   1656   1664   1672   1680   1795   1803
            1811   1819   1934   1942   1950   1958   2073   2081   2089   2097   2212   2220   2228   2236
            2351   2359   2367   2375   2490   2498   2506   2514
.MSKCN       111    113    115    117    119    121    123    125    127    129    131    133    135    137
             139    141    143    306    310    314    318    322    326    330    334    339    343    347
             351    355    359    363    367    372    376    380    384    388    392    396    400    405
             409    413    417    421    425    429    433    445    449    453    457    461    465    469
             473    478    482    486    490    494    498    502    506    511    515    519    523    527
             531    535    539    544    548    552    556    560    564    568    572    584    588    592
             596    600    604    608    612    617    621    625    629    633    637    641    645    650
```

```
           654   658   662   666   670   674   678   683   687   691   695   699   703   707
           711   723   727   731   735   739   743   747   751   756   760   764   768   772
           776   780   784   789   793   797   801   805   809   813   817   822   826   830
           834   838   842   846   850   862   866   870   874   878   882   886   890   895
           899   903   907   911   915   919   923   928   932   936   940   944   948   952
           956   961   965   969   973   977   981   985   989  1001  1005  1009  1013  1017
          1021  1025  1029  1034  1038  1042  1046  1050  1054  1058  1062  1067  1071  1075
          1079  1083  1087  1091  1095  1100  1104  1108  1112  1116  1120  1124  1128  1140
          1144  1148  1152  1156  1160  1164  1168  1173  1177  1181  1185  1189  1193  1197
          1201  1206  1210  1214  1218  1222  1226  1230  1234  1239  1243  1247  1251  1255
          1259  1263  1267  1279  1283  1287  1291  1295  1299  1303  1307  1312  1316  1320
          1324  1328  1332  1336  1340  1345  1349  1353  1357  1361  1365  1369  1373  1378
          1382  1386  1390  1394  1398  1402  1406  1418  1422  1426  1430  1434  1438  1442
          1446  1451  1455  1459  1463  1467  1471  1475  1479  1484  1488  1492  1496  1500
          1504  1508  1512  1517  1521  1525  1529  1533  1537  1541  1545  1557  1561  1565
          1569  1573  1577  1581  1585  1590  1594  1598  1602  1606  1610  1614  1618  1623
          1627  1631  1635  1639  1643  1647  1651  1656  1660  1664  1668  1672  1676  1680
          1684  1696  1700  1704  1708  1712  1716  1720  1724  1729  1733  1737  1741  1745
          1749  1753  1757  1762  1766  1770  1774  1778  1782  1786  1790  1795  1799  1803
          1807  1811  1815  1819  1823  1835  1839  1843  1847  1851  1855  1859  1863  1868
          1872  1876  1880  1884  1888  1892  1896  1901  1905  1909  1913  1917  1921  1925
          1929  1934  1938  1942  1946  1950  1954  1958  1962  1974  1978  1982  1986  1990
          1994  1998  2002  2007  2011  2015  2019  2023  2027  2031  2035  2040  2044  2048
          2052  2056  2060  2064  2068  2073  2077  2081  2085  2089  2093  2097  2101  2113
          2117  2121  2125  2129  2133  2137  2141  2146  2150  2154  2158  2162  2166  2170
          2174  2179  2183  2187  2191  2195  2199  2203  2207  2212  2216  2220  2224  2228
          2232  2236  2240  2252  2256  2260  2264  2268  2272  2276  2280  2285  2289  2293
          2297  2301  2305  2309  2313  2318  2322  2326  2330  2334  2338  2342  2346  2351
          2355  2359  2363  2367  2371  2375  2379  2391  2395  2399  2403  2407  2411  2415
          2419  2424  2428  2432  2436  2440  2444  2448  2452  2457  2461  2465  2469  2473
          2477  2481  2485  2490  2494  2498  2502  2506  2510  2514  2518
.MSMIN     310   363   376   429   449   502   515   568   588   641   654   707   727   780
           793   846   866   919   932   985  1005  1058  1071  1124  1144  1197  1210  1263
          1283  1336  1349  1402  1422  1475  1488  1541  1561  1614  1627  1680  1700  1753
          1766  1819  1839  1892  1905  1958  1978  2031  2044  2097  2117  2170  2183  2236
          2256  2309  2322  2375  2395  2448  2461  2514
.MXNOR     334   339   400   405   473   478   539   544   612   617   678   683   751   756
           817   822   890   895   956   961  1029  1034  1095  1100  1168  1173  1234  1239
          1307  1312  1373  1378  1446  1451  1512  1517  1585  1590  1651  1656  1724  1729
          1790  1795  1863  1868  1929  1934  2002  2007  2068  2073  2141  2146  2207  2212
          2280  2285  2346  2351  2419  2424  2485  2490
.MXOR      330   343   396   409   469   482   535   548   608   621   674   687   747   760
           813   826   886   899   952   965  1025  1038  1091  1104  1164  1177  1230  1243
          1303  1316  1369  1382  1442  1455  1508  1521  1581  1594  1647  1660  1720  1733
          1786  1799  1859  1872  1925  1938  1998  2011  2064  2077  2137  2150  2203  2216
          2276  2289  2342  2355  2415  2428  2481  2494
.RA        159   168   177   186   195   204   213   222   231   240   249   258   267   276
           285   303   306   308   310   312   314   316   318   320   322   324   326   328
           330   332   334   336   339   341   343   345   347   349   351   353   355   357
           359   361   363   365   367   369   372   374   376   378   380   382   384   386
           388   390   392   394   396   398   400   402   405   407   409   411   413   415
           417   419   421   423   425   427   429   431   433   435   442   445   447   449
           451   453   455   457   459   461   463   465   467   469   471   473   475   478
```

480	482	484	486	488	490	492	494	496	498	500	502	504	506
508	511	513	515	517	519	521	523	525	527	529	531	533	535
537	539	541	544	546	548	550	552	554	556	558	560	562	564
566	568	570	572	574	581	584	586	588	590	592	594	596	598
600	602	604	606	608	610	612	614	617	619	621	623	625	627
629	631	633	635	637	639	641	643	645	647	650	652	654	656
658	660	662	664	666	668	670	672	674	676	678	680	683	685
687	689	691	693	695	697	699	701	703	705	707	709	711	713
720	723	725	727	729	731	733	735	737	739	741	743	745	747
749	751	753	756	758	760	762	764	766	768	770	772	774	776
778	780	782	784	786	789	791	793	795	797	799	801	803	805
807	809	811	813	815	817	819	822	824	826	828	830	832	834
836	838	840	842	844	846	848	850	852	859	862	864	866	868
870	872	874	876	878	880	882	884	886	888	890	892	895	897
899	901	903	905	907	909	911	913	915	917	919	921	923	925
928	930	932	934	936	938	940	942	944	946	948	950	952	954
956	958	961	963	965	967	969	971	973	975	977	979	981	983
985	987	989	991	998	1001	1003	1005	1007	1009	1011	1013	1015	1017
1019	1021	1023	1025	1027	1029	1031	1034	1036	1038	1040	1042	1044	1046
1048	1050	1052	1054	1056	1058	1060	1062	1064	1067	1069	1071	1073	1075
1077	1079	1081	1083	1085	1087	1089	1091	1093	1095	1097	1100	1102	1104
1106	1108	1110	1112	1114	1116	1118	1120	1122	1124	1126	1128	1130	1137
1140	1142	1144	1146	1148	1150	1152	1154	1156	1158	1160	1162	1164	1166
1168	1170	1173	1175	1177	1179	1181	1183	1185	1187	1189	1191	1193	1195
1197	1199	1201	1203	1206	1208	1210	1212	1214	1216	1218	1220	1222	1224
1226	1228	1230	1232	1234	1236	1239	1241	1243	1245	1247	1249	1251	1253
1255	1257	1259	1261	1263	1265	1267	1269	1276	1279	1281	1283	1285	1287
1289	1291	1293	1295	1297	1299	1301	1303	1305	1307	1309	1312	1314	1316
1318	1320	1322	1324	1326	1328	1330	1332	1334	1336	1338	1340	1342	1345
1347	1349	1351	1353	1355	1357	1359	1361	1363	1365	1367	1369	1371	1373
1375	1378	1380	1382	1384	1386	1388	1390	1392	1394	1396	1398	1400	1402
1404	1406	1408	1415	1418	1420	1422	1424	1426	1428	1430	1432	1434	1436
1438	1440	1442	1444	1446	1448	1451	1453	1455	1457	1459	1461	1463	1465
1467	1469	1471	1473	1475	1477	1479	1481	1484	1486	1488	1490	1492	1494
1496	1498	1500	1502	1504	1506	1508	1510	1512	1514	1517	1519	1521	1523
1525	1527	1529	1531	1533	1535	1537	1539	1541	1543	1545	1547	1554	1557
1559	1561	1563	1565	1567	1569	1571	1573	1575	1577	1579	1581	1583	1585
1587	1590	1592	1594	1596	1598	1600	1602	1604	1606	1608	1610	1612	1614
1616	1618	1620	1623	1625	1627	1629	1631	1633	1635	1637	1639	1641	1643
1645	1647	1649	1651	1653	1656	1658	1660	1662	1664	1666	1668	1670	1672
1674	1676	1678	1680	1682	1684	1686	1693	1696	1698	1700	1702	1704	1706
1708	1710	1712	1714	1716	1718	1720	1722	1724	1726	1729	1731	1733	1735
1737	1739	1741	1743	1745	1747	1749	1751	1753	1755	1757	1759	1762	1764
1766	1768	1770	1772	1774	1776	1778	1780	1782	1784	1786	1788	1790	1792
1795	1797	1799	1801	1803	1805	1807	1809	1811	1813	1815	1817	1819	1821
1823	1825	1832	1835	1837	1839	1841	1843	1845	1847	1849	1851	1853	1855
1857	1859	1861	1863	1865	1868	1870	1872	1874	1876	1878	1880	1882	1684
1886	1888	1890	1892	1894	1896	1898	1901	1903	1905	1907	1909	1911	1913
1915	1917	1919	1921	1923	1925	1927	1929	1931	1934	1936	1938	1940	1942
1944	1946	1948	1950	1952	1954	1956	1958	1960	1962	1964	1971	1974	1976
1978	1980	1982	1984	1986	1988	1990	1992	1994	1996	1998	2000	2002	2004
2007	2009	2011	2013	2015	2017	2019	2021	2023	2025	2027	2029	2031	2033
2035	2037	2040	2042	2044	2046	2048	2050	2052	2054	2056	2058	2060	2062

```
        2064  2066  2068  2070  2073  2075  2077  2079  2081  2083  2085  2087  2089  2091   SEQ 1278
        2093  2095  2097  2099  2101  2103  2110  2113  2115  2117  2119  2121  2123  2125
        2127  2129  2131  2133  2135  2137  2139  2141  2143  2146  2148  2150  2152  2154
        2156  2158  2160  2162  2164  2166  2168  2170  2172  2174  2176  2179  2181  2183
        2185  2187  2189  2191  2193  2195  2197  2199  2201  2203  2205  2207  2209  2212
        2214  2216  2218  2220  2222  2224  2226  2228  2230  2232  2234  2236  2238  2240
        2242  2249  2252  2254  2256  2258  2260  2262  2264  2266  2268  2270  2272  2274
        2276  2278  2280  2282  2285  2287  2289  2291  2293  2295  2297  2299  2301  2303
        2305  2307  2309  2311  2313  2315  2318  2320  2322  2324  2326  2328  2330  2332
        2334  2336  2338  2340  2342  2344  2346  2348  2351  2353  2355  2357  2359  2361
        2363  2365  2367  2369  2371  2373  2375  2377  2379  2381  2388  2391  2393  2395
        2397  2399  2401  2403  2405  2407  2409  2411  2413  2415  2417  2419  2421  2424
        2426  2428  2430  2432  2434  2436  2438  2440  2442  2444  2446  2448  2450  2452
        2454  2457  2459  2461  2463  2465  2467  2469  2471  2473  2475  2477  2479  2481
        2483  2485  2487  2490  2492  2494  2496  2498  2500  2502  2504  2506  2508  2510
        2512  2514  2516  2518  2520
.RAND    303   322   351   388   417   442   461   490   527   556   581   600   629   666
         695   720   739   768   805   834   859   878   907   944   973   998  1017  1046
        1083  1112  1137  1156  1185  1222  1251  1276  1295  1324  1361  1390  1415  1434
        1463  1500  1529  1554  1573  1602  1639  1668  1693  1712  1741  1778  1807  1832
        1851  1880  1917  1946  1971  1990  2019  2056  2085  2110  2129  2158  2195  2224
        2249  2268  2297  2334  2363  2388  2407  2436  2473  2502
.RB      111   113   115   117   119   121   123   125   127   129   131   133   135   137
         139   141   143   303   306   310   314   318   322   326   330   334   339   343
         347   351   355   359   363   367   372   376   380   384   388   392   396   400
         405   409   413   417   421   425   429   433   442   445   449   453   457   461
         465   469   473   478   482   486   490   494   498   502   506   511   515   519
         523   527   531   535   539   544   548   552   556   560   564   568   572   581
         584   588   592   596   600   604   608   612   617   621   625   629   633   637
         641   645   650   654   658   662   666   670   674   678   683   687   691   695
         699   703   707   711   720   723   727   731   735   739   743   747   751   756
         760   764   768   772   776   780   784   789   793   797   801   805   809   813
         817   822   826   830   834   838   842   846   850   859   862   866   870   874
         878   882   886   890   895   899   903   907   911   915   919   923   928   932
         936   940   944   948   952   956   961   965   969   973   977   981   985   989
         998  1001  1005  1009  1013  1017  1021  1025  1029  1034  1038  1042  1046  1050
        1054  1058  1062  1067  1071  1075  1079  1083  1087  1091  1095  1100  1104  1108
        1112  1116  1120  1124  1128  1137  1140  1144  1148  1152  1156  1160  1164  1168
        1173  1177  1181  1185  1189  1193  1197  1201  1206  1210  1214  1218  1222  1226
        1230  1234  1239  1243  1247  1251  1255  1259  1263  1267  1276  1279  1283  1287
        1291  1295  1299  1303  1307  1312  1316  1320  1324  1328  1332  1336  1340  1345
        1349  1353  1357  1361  1365  1369  1373  1378  1382  1386  1390  1394  1398  1402
        1406  1415  1418  1422  1426  1430  1434  1438  1442  1446  1451  1455  1459  1463
        1467  1471  1475  1479  1484  1488  1492  1496  1500  1504  1508  1512  1517  1521
        1525  1529  1533  1537  1541  1545  1554  1557  1561  1565  1569  1573  1577  1581
        1585  1590  1594  1598  1602  1606  1610  1614  1618  1623  1627  1631  1635  1639
        1643  1647  1651  1656  1660  1664  1668  1672  1676  1680  1684  1693  1696  1700
        1704  1708  1712  1716  1720  1724  1729  1733  1737  1741  1745  1749  1753  1757
        1762  1766  1770  1774  1778  1782  1786  1790  1795  1799  1803  1807  1811  1815
        1819  1823  1832  1835  1839  1843  1847  1851  1855  1859  1863  1868  1872  1876
        1880  1884  1888  1892  1896  1901  1905  1909  1913  1917  1921  1925  1929  1934
        1938  1942  1946  1950  1954  1958  1962  1971  1974  1978  1982  1986  1990  1994
        1998  2002  2007  2011  2015  2019  2023  2027  2031  2035  2040  2044  2048  2052
```

```
            2056    2060    2064    2068    2073    2077    2081    2085    2089    2093    2097    2101    2110    2113      SEQ 1279
            2117    2121    2125    2129    2133    2137    2141    2146    2150    2154    2158    2162    2166    2170
            2174    2179    2183    2187    2191    2195    2199    2203    2207    2212    2216    2220    2224    2228
            2232    2236    2240    2249    2252    2256    2260    2264    2268    2272    2276    2280    2285    2289
            2293    2297    2301    2305    2309    2313    2318    2322    2326    2330    2334    2338    2342    2346
            2351    2355    2359    2363    2367    2371    2375    2379    2388    2391    2395    2399    2403    2407
            2411    2415    2419    2424    2428    2432    2436    2440    2444    2448    2452    2457    2461    2465
            2469    2473    2477    2481    2485    2490    2494    2498    2502    2506    2510    2514    2518
.RBAD        152     161     170     179     188     197     206     215     224     233     242     251     260     269
             278     287     294     296
.RCCFZ       150     159     168     177     186     195     204     213     222     231     240     249     258     267
             276     285
.RCENA       150     159     168     177     186     195     204     213     222     231     240     249     258     267
             276     285
.RCJS        301     308     312     316     320     324     328     332     336     341     345     349     353     357
             361     365     369     374     378     382     386     390     394     398     402     407     411     415
             419     423     427     431     435     440     447     451     455     459     463     467     471     475
             480     484     488     492     496     500     504     508     513     517     521     525     529     533
             537     541     546     550     554     558     562     566     570     574     579     586     590     594
             598     602     606     610     614     619     623     627     631     635     639     643     647     652
             656     660     664     668     672     676     680     685     689     693     697     701     705     709
             713     718     725     729     733     737     741     745     749     753     758     762     766     770
             774     778     782     786     791     795     799     803     807     811     815     819     824     828
             832     836     840     844     848     852     857     864     868     872     876     880     884     888
             892     897     901     905     909     913     917     921     925     930     934     938     942     946
             950     954     958     963     967     971     975     979     983     987     991     996    1003    1007
            1011    1015    1019    1023    1027    1031    1036    1040    1044    1048    1052    1056    1060    1064
            1069    1073    1077    1081    1085    1089    1093    1097    1102    1106    1110    1114    1118    1122
            1126    1130    1135    1142    1146    1150    1154    1158    1162    1166    1170    1175    1179    1183
            1187    1191    1195    1199    1203    1208    1212    1216    1220    1224    1228    1232    1236    1241
            1245    1249    1253    1257    1261    1265    1269    1274    1281    1285    1289    1293    1297    1301
            1305    1309    1314    1318    1322    1326    1330    1334    1338    1342    1347    1351    1355    1359
            1363    1367    1371    1375    1380    1384    1388    1392    1396    1400    1404    1408    1413    1420
            1424    1428    1432    1436    1440    1444    1448    1453    1457    1461    1465    1469    1473    1477
            1481    1486    1490    1494    1498    1502    1506    1510    1514    1519    1523    1527    1531    1535
            1539    1543    1547    1552    1559    1563    1567    1571    1575    1579    1583    1587    1592    1596
            1600    1604    1608    1612    1616    1620    1625    1629    1633    1637    1641    1645    1649    1653
            1658    1662    1666    1670    1674    1678    1682    1686    1691    1698    1702    1706    1710    1714
            1718    1722    1726    1731    1735    1739    1743    1747    1751    1755    1759    1764    1768    1772
            1776    1780    1784    1788    1792    1797    1801    1805    1809    1813    1817    1821    1825    1830
            1837    1841    1845    1849    1853    1857    1861    1865    1870    1874    1878    1882    1886    1890
            1894    1898    1903    1907    1911    1915    1919    1923    1927    1931    1936    1940    1944    1948
            1952    1956    1960    1964    1969    1976    1980    1984    1988    1992    1996    2000    2004    2009
            2013    2017    2021    2025    2029    2033    2037    2042    2046    2050    2054    2058    2062    2066
            2070    2075    2079    2083    2087    2091    2095    2099    2103    2108    2115    2119    2123    2127
            2131    2135    2139    2143    2148    2152    2156    2160    2164    2168    2172    2176    2181    2185
            2189    2193    2197    2201    2205    2209    2214    2218    2222    2226    2230    2234    2238    2242
            2247    2254    2258    2262    2266    2270    2274    2278    2282    2287    2291    2295    2299    2303
            2307    2311    2315    2320    2324    2328    2332    2336    2340    2344    2348    2353    2357    2361
            2365    2369    2373    2377    2381    2386    2393    2397    2401    2405    2409    2413    2417    2421
            2426    2430    2434    2438    2442    2446    2450    2454    2459    2463    2467    2471    2475    2479
            2483    2487    2492    2496    2500    2504    2508    2512    2516    2520
.RCJV        150     159     168     177     186     195     204     213     222     231     240     249     258     267
```

```
           276   285
.RCONT     111   113   115   117   119   121   123   125   127   129   131   133   135   137     SEQ 1280
           139   141   143   303   306   310   314   318   322   326   330   334   339   343
           347   351   355   359   363   367   372   376   380   384   388   392   396   400
           405   409   413   417   421   425   429   433   442   445   449   453   457   461
           465   469   473   478   482   486   490   494   498   502   506   511   515   519
           523   527   531   535   539   544   548   552   556   560   564   568   572   581
           584   588   592   596   600   604   608   612   617   621   625   629   633   637
           641   645   650   654   658   662   666   670   674   678   683   687   691   695
           699   703   707   711   720   723   727   731   735   739   743   747   751   756
           760   764   768   772   776   780   784   789   793   797   801   805   809   813
           817   822   826   830   834   838   842   846   850   859   862   866   870   874
           878   882   886   890   895   899   903   907   911   915   919   923   928   932
           936   940   944   948   952   956   961   965   969   973   977   981   985   989
           998  1001  1005  1009  1013  1017  1021  1025  1029  1034  1038  1042  1046  1050
          1054  1058  1062  1067  1071  1075  1079  1083  1087  1091  1095  1100  1104  1108
          1112  1116  1120  1124  1128  1137  1140  1144  1148  1152  1156  1160  1164  1168
          1173  1177  1181  1185  1189  1193  1197  1201  1206  1210  1214  1218  1222  1226
          1230  1234  1239  1243  1247  1251  1255  1259  1263  1267  1276  1279  1283  1287
          1291  1295  1299  1303  1307  1312  1316  1320  1324  1328  1332  1336  1340  1345
          1349  1353  1357  1361  1365  1369  1373  1378  1382  1386  1390  1394  1398  1402
          1406  1415  1418  1422  1426  1430  1434  1438  1442  1446  1451  1455  1459  1463
          1467  1471  1475  1479  1484  1488  1492  1496  1500  1504  1508  1512  1517  1521
          1525  1529  1533  1537  1541  1545  1554  1557  1561  1565  1569  1573  1577  1581
          1585  1590  1594  1598  1602  1606  1610  1614  1618  1623  1627  1631  1635  1639
          1643  1647  1651  1656  1660  1664  1668  1672  1676  1680  1684  1693  1696  1700
          1704  1708  1712  1716  1720  1724  1729  1733  1737  1741  1745  1749  1753  1757
          1762  1766  1770  1774  1778  1782  1786  1790  1795  1799  1803  1807  1811  1815
          1819  1823  1832  1835  1839  1843  1847  1851  1855  1859  1863  1868  1872  1876
          1880  1884  1888  1892  1896  1901  1905  1909  1913  1917  1921  1925  1929  1934
          1938  1942  1946  1950  1954  1958  1962  1971  1974  1978  1982  1986  1990  1994
          1998  2002  2007  2011  2015  2019  2023  2027  2031  2035  2040  2044  2048  2052
          2056  2060  2064  2068  2073  2077  2081  2085  2089  2093  2097  2101  2110  2113
          2117  2121  2125  2129  2133  2137  2141  2146  2150  2154  2158  2162  2166  2170
          2174  2179  2183  2187  2191  2195  2199  2203  2207  2212  2216  2220  2224  2228
          2232  2236  2240  2249  2252  2256  2260  2264  2268  2272  2276  2280  2285  2289
          2293  2297  2301  2305  2309  2313  2318  2322  2326  2330  2334  2338  2342  2346
          2351  2355  2359  2363  2367  2371  2375  2379  2388  2391  2395  2399  2403  2407
          2411  2415  2419  2424  2428  2432  2436  2440  2444  2448  2452  2457  2461  2465
          2469  2473  2477  2481  2485  2490  2494  2498  2502  2506  2510  2514  2518
.RCRTN     145   154   163   172   181   190   199   208   217   226   235   244   253   262
           271   280   289
.RD        111   113   115   117   119   121   123   125   127   129   131   133   135   137
           139   141   143   145   150   159   168   177   186   195   204   213   222   231
           240   249   258   267   276   285   303   306   308   310   312   314   316   318
           320   322   324   326   328   330   332   334   336   339   341   343   345   347
           349   351   353   355   357   359   361   363   365   367   369   372   374   376
           378   380   382   384   386   388   390   392   394   396   398   400   402   405
           407   409   411   413   415   417   419   421   423   425   427   429   431   433
           435   442   445   447   449   451   453   455   457   459   461   463   465   467
           469   471   473   475   478   480   482   484   486   488   490   492   494   496
           498   500   502   504   506   508   511   513   515   517   519   521   523   525
           527   529   531   533   535   537   539   541   544   546   548   550   552   554
```

556	558	560	562	564	566	568	570	572	574	581	584	586	588
590	592	594	596	598	600	602	604	606	608	610	612	614	617
619	621	623	625	627	629	631	633	635	637	639	641	643	645
647	650	652	654	656	658	660	662	664	666	668	670	672	674
676	678	680	683	685	687	689	691	693	695	697	699	701	703
705	707	709	711	713	720	723	725	727	729	731	733	735	737
739	741	743	745	747	749	751	753	756	758	760	762	764	766
768	770	772	774	776	778	780	782	784	786	789	791	793	795
797	799	801	803	805	807	809	811	813	815	817	819	822	824
826	828	830	832	834	836	838	840	842	844	846	848	850	852
859	862	864	866	868	870	872	874	876	878	880	882	884	886
888	890	892	895	897	899	901	903	905	907	909	911	913	915
917	919	921	923	925	928	930	932	934	936	938	940	942	944
946	948	950	952	954	956	958	961	963	965	967	969	971	973
975	977	979	981	983	985	987	989	991	998	1001	1003	1005	1007
1009	1011	1013	1015	1017	1019	1021	1023	1025	1027	1029	1031	1034	1036
1038	1040	1042	1044	1046	1048	1050	1052	1054	1056	1058	1060	1062	1064
1067	1069	1071	1073	1075	1077	1079	1081	1083	1085	1087	1089	1091	1093
1095	1097	1100	1102	1104	1106	1108	1110	1112	1114	1116	1118	1120	1122
1124	1126	1128	1130	1137	1140	1142	1144	1146	1148	1150	1152	1154	1156
1158	1160	1162	1164	1166	1168	1170	1173	1175	1177	1179	1181	1183	1185
1187	1189	1191	1193	1195	1197	1199	1201	1203	1206	1208	1210	1212	1214
1216	1218	1220	1222	1224	1226	1228	1230	1232	1234	1236	1239	1241	1243
1245	1247	1249	1251	1253	1255	1257	1259	1261	1263	1265	1267	1269	1276
1279	1281	1283	1285	1287	1289	1291	1293	1295	1297	1299	1301	1303	1305
1307	1309	1312	1314	1316	1318	1320	1322	1324	1326	1328	1330	1332	1334
1336	1338	1340	1342	1345	1347	1349	1351	1353	1355	1357	1359	1361	1363
1365	1367	1369	1371	1373	1375	1378	1380	1382	1384	1386	1388	1390	1392
1394	1396	1398	1400	1402	1404	1406	1408	1415	1418	1420	1422	1424	1426
1428	1430	1432	1434	1436	1438	1440	1442	1444	1446	1448	1451	1453	1455
1457	1459	1461	1463	1465	1467	1469	1471	1473	1475	1477	1479	1481	1484
1486	1488	1490	1492	1494	1496	1498	1500	1502	1504	1506	1508	1510	1512
1514	1517	1519	1521	1523	1525	1527	1529	1531	1533	1535	1537	1539	1541
1543	1545	1547	1554	1557	1559	1561	1563	1565	1567	1569	1571	1573	1575
1577	1579	1581	1583	1585	1587	1590	1592	1594	1596	1598	1600	1602	1604
1606	1608	1610	1612	1614	1616	1618	1620	1623	1625	1627	1629	1631	1633
1635	1637	1639	1641	1643	1645	1647	1649	1651	1653	1656	1658	1660	1662
1664	1666	1668	1670	1672	1674	1676	1678	1680	1682	1684	1686	1693	1696
1698	1700	1702	1704	1706	1708	1710	1712	1714	1716	1718	1720	1722	1724
1726	1729	1731	1733	1735	1737	1739	1741	1743	1745	1747	1749	1751	1753
1755	1757	1759	1762	1764	1766	1768	1770	1772	1774	1776	1778	1780	1782
1784	1786	1788	1790	1792	1795	1797	1799	1801	1803	1805	1807	1809	1811
1813	1815	1817	1819	1821	1823	1825	1832	1835	1837	1839	1841	1843	1845
1847	1849	1851	1853	1855	1857	1859	1861	1863	1865	1868	1870	1872	1874
1876	1878	1880	1882	1884	1886	1888	1890	1892	1894	1896	1898	1901	1903
1905	1907	1909	1911	1913	1915	1917	1919	1921	1923	1925	1927	1929	1931
1934	1936	1938	1940	1942	1944	1946	1948	1950	1952	1954	1956	1958	1960
1962	1964	1971	1974	1976	1978	1980	1982	1984	1986	1988	1990	1992	1994
1996	1998	2000	2002	2004	2007	2009	2011	2013	2015	2017	2019	2021	2023
2025	2027	2029	2031	2033	2035	2037	2040	2042	2044	2046	2048	2050	2052
2054	2056	2058	2060	2062	2064	2066	2068	2070	2073	2075	2077	2079	2081
2083	2085	2087	2089	2091	2093	2095	2097	2099	2101	2103	2110	2113	2115
2117	2119	2121	2123	2125	2127	2129	2131	2133	2135	2137	2139	2141	2143

1269

```
      2146  2148  2150  2152  2154  2156  2158  2160  2162  2164  2166  2168  2170  2172   SEQ 1282
      2174  2176  2179  2181  2183  2185  2187  2189  2191  2193  2195  2197  2199  2201
      2203  2205  2207  2209  2212  2214  2216  2218  2220  2222  2224  2226  2228  2230
      2232  2234  2236  2238  2240  2242  2249  2252  2254  2256  2258  2260  2262  2264
      2266  2268  2270  2272  2274  2276  2278  2280  2282  2285  2287  2289  2291  2293
      2295  2297  2299  2301  2303  2305  2307  2309  2311  2313  2315  2318  2320  2322
      2324  2326  2328  2330  2332  2334  2336  2338  2340  2342  2344  2346  2348  2351
      2353  2355  2357  2359  2361  2363  2365  2367  2369  2371  2373  2375  2377  2379
      2381  2388  2391  2393  2395  2397  2399  2401  2403  2405  2407  2409  2411  2413
      2415  2417  2419  2421  2424  2426  2428  2430  2432  2434  2436  2438  2440  2442
      2444  2446  2448  2450  2452  2454  2457  2459  2461  2463  2465  2467  2469  2471
      2473  2475  2477  2479  2481  2483  2485  2487  2490  2492  2494  2496  2498  2500
      2502  2504  2506  2508  2510  2512  2514  2516  2518  2520
.RJ    103   150   152   159   161   168   170   177   179   186   188   195   197   204
       206   213   215   222   224   231   233   240   242   249   251   258   260   267
       269   276   278   285   287   294   296   301   308   312   316   320   324   328
       332   336   341   345   349   353   357   361   365   369   374   378   382   386
       390   394   398   402   407   411   415   419   423   427   431   435   440   447
       451   455   459   463   467   471   475   480   484   488   492   496   500   504
       508   513   517   521   525   529   533   537   541   546   550   554   558   562
       566   570   574   579   586   590   594   598   602   606   610   614   619   623
       627   631   635   639   643   647   652   656   660   664   668   672   676   680
       685   689   693   697   701   705   709   713   718   725   729   733   737   741
       745   749   753   758   762   766   770   774   778   782   786   791   795   799
       803   807   811   815   819   824   828   832   836   840   844   848   852   857
       864   868   872   876   880   884   888   892   897   901   905   909   913   917
       921   925   930   934   938   942   946   950   954   958   963   967   971   975
       979   983   987   991   996  1003  1007  1011  1015  1019  1023  1027  1031  1036
      1040  1044  1048  1052  1056  1060  1064  1069  1073  1077  1081  1085  1089  1093
      1097  1102  1106  1110  1114  1118  1122  1126  1130  1135  1142  1146  1150  1154
      1158  1162  1166  1170  1175  1179  1183  1187  1191  1195  1199  1203  1208  1212
      1216  1220  1224  1228  1232  1236  1241  1245  1249  1253  1257  1261  1265  1269
      1274  1281  1285  1289  1293  1297  1301  1305  1309  1314  1318  1322  1326  1330
      1334  1338  1342  1347  1351  1355  1359  1363  1367  1371  1375  1380  1384  1388
      1392  1396  1400  1404  1408  1413  1420  1424  1428  1432  1436  1440  1444  1448
      1453  1457  1461  1465  1469  1473  1477  1481  1486  1490  1494  1498  1502  1506
      1510  1514  1519  1523  1527  1531  1535  1539  1543  1547  1552  1559  1563  1567
      1571  1575  1579  1583  1587  1592  1596  1600  1604  1608  1612  1616  1620  1625
      1629  1633  1637  1641  1645  1649  1653  1658  1662  1666  1670  1674  1678  1682
      1686  1691  1698  1702  1706  1710  1714  1718  1722  1726  1731  1735  1739  1743
      1747  1751  1755  1759  1764  1768  1772  1776  1780  1784  1788  1792  1797  1801
      1805  1809  1813  1817  1821  1825  1830  1837  1841  1845  1849  1853  1857  1861
      1865  1870  1874  1878  1882  1886  1890  1894  1898  1903  1907  1911  1915  1919
      1923  1927  1931  1936  1940  1944  1948  1952  1956  1960  1964  1969  1976  1980
      1984  1988  1992  1996  2000  2004  2009  2013  2017  2021  2025  2029  2033  2037
      2042  2046  2050  2054  2058  2062  2066  2070  2075  2079  2083  2087  2091  2095
      2099  2103  2108  2115  2119  2123  2127  2131  2135  2139  2143  2148  2152  2156
      2160  2164  2168  2172  2176  2181  2185  2189  2193  2197  2201  2205  2209  2214
      2218  2222  2226  2230  2234  2238  2242  2247  2254  2258  2262  2266  2270  2274
      2278  2282  2287  2291  2295  2299  2303  2307  2311  2315  2320  2324  2328  2332
      2336  2340  2344  2348  2353  2357  2361  2365  2369  2373  2377  2381  2386  2393
      2397  2401  2405  2409  2413  2417  2421  2426  2430  2434  2438  2442  2446  2450
      2454  2459  2463  2467  2471  2475  2479  2483  2487  2492  2496  2500  2504  2508
```

MWOR

```
        2512  2516  2520  2522
.RJMAP   103   152   161   170   179   188   197   206   215   224   233   242   251   260     SEQ 1283
         269   278   287   294   296  2522
.RJZ     105
.RMGC    111   113   115   117   119   121   123   125   127   129   131   133   135   137
         139   141   143   306   310   314   318   322   326   330   334   339   343   347
         351   355   359   363   367   372   376   380   384   388   392   396   400   405
         409   413   417   421   425   429   433   445   449   453   457   461   465   469
         473   478   482   486   490   494   498   502   506   511   515   519   523   527
         531   535   539   544   548   552   556   560   564   568   572   584   588   592
         596   600   604   608   612   617   621   625   629   633   637   641   645   650
         654   658   662   666   670   674   678   683   687   691   695   699   703   707
         711   723   727   731   735   739   743   747   751   756   760   764   768   772
         776   780   784   789   793   797   801   805   809   813   817   822   826   830
         834   838   842   846   850   862   866   870   874   878   882   886   890   895
         899   903   907   911   915   919   923   928   932   936   940   944   948   952
         956   961   965   969   973   977   981   985   989  1001  1005  1009  1013  1017
        1021  1025  1029  1034  1038  1042  1046  1050  1054  1058  1062  1067  1071  1075
        1079  1083  1087  1091  1095  1100  1104  1108  1112  1116  1120  1124  1128  1140
        1144  1148  1152  1156  1160  1164  1168  1173  1177  1181  1185  1189  1193  1197
        1201  1206  1210  1214  1218  1222  1226  1230  1234  1239  1243  1247  1251  1255
        1259  1263  1267  1279  1283  1287  1291  1295  1299  1303  1307  1312  1316  1320
        1324  1328  1332  1336  1340  1345  1349  1353  1357  1361  1365  1369  1373  1378
        1382  1386  1390  1394  1398  1402  1406  1418  1422  1426  1430  1434  1438  1442
        1446  1451  1455  1459  1463  1467  1471  1475  1479  1484  1488  1492  1496  1500
        1504  1508  1512  1517  1521  1525  1529  1533  1537  1541  1545  1557  1561  1565
        1569  1573  1577  1581  1585  1590  1594  1598  1602  1606  1610  1614  1618  1623
        1627  1631  1635  1639  1643  1647  1651  1656  1660  1664  1668  1672  1676  1680
        1684  1696  1700  1704  1708  1712  1716  1720  1724  1729  1733  1737  1741  1745
        1749  1753  1757  1762  1766  1770  1774  1778  1782  1786  1790  1795  1799  1803
        1807  1811  1815  1819  1823  1835  1839  1843  1847  1851  1855  1859  1863  1868
        1872  1876  1880  1884  1888  1892  1896  1901  1905  1909  1913  1917  1921  1925
        1929  1934  1938  1942  1946  1950  1954  1958  1962  1974  1978  1982  1986  1990
        1994  1998  2002  2007  2011  2015  2019  2023  2027  2031  2035  2040  2044  2048
        2052  2056  2060  2064  2068  2073  2077  2081  2085  2089  2093  2097  2101  2113
        2117  2121  2125  2129  2133  2137  2141  2146  2150  2154  2158  2162  2166  2170
        2174  2179  2183  2187  2191  2195  2199  2203  2207  2212  2216  2220  2224  2228
        2232  2236  2240  2252  2256  2260  2264  2268  2272  2276  2280  2285  2289  2293
        2297  2301  2305  2309  2313  2318  2322  2326  2330  2334  2338  2342  2346  2351
        2355  2359  2363  2367  2371  2375  2379  2391  2395  2399  2403  2407  2411  2415
        2419  2424  2428  2432  2436  2440  2444  2448  2452  2457  2461  2465  2469  2473
        2477  2481  2485  2490  2494  2498  2502  2506  2510  2514  2518
.RNAND   326   347   392   413   465   486   531   552   604   625   670   691   743   764
         809   830   882   903   948   969  1021  1042  1087  1108  1160  1181  1226  1247
        1299  1320  1365  1386  1438  1459  1504  1525  1577  1598  1643  1664  1716  1737
        1782  1803  1855  1876  1921  1942  1994  2015  2060  2081  2133  2154  2199  2220
        2272  2293  2338  2359  2411  2432  2477  2498
.ROR     111   113   115   117   119   121   123   125   127   129   131   133   135   137
         139   141   143   150   159   168   177   186   195   204   213   222   231   240
         249   258   267   276   285   308   312   316   318   320   324   328   332   336
         341   345   349   353   355   357   361   365   369   374   378   382   384   386
         390   394   398   402   407   411   415   419   421   423   427   431   435   447
         451   455   457   459   463   467   471   475   480   484   488   492   494   496
```

```
          500   504   508   513   517   521   523   525   529   533   537   541   546   550    SEQ 1284
          554   558   560   562   566   570   574   586   590   594   596   598   602   606
          610   614   619   623   627   631   633   635   639   643   647   652   656   660
          662   664   668   672   676   680   685   689   693   697   699   701   705   709
          713   725   729   733   735   737   741   745   749   753   758   762   766   770
          772   774   778   782   786   791   795   799   801   803   807   811   815   819
          824   828   832   836   838   840   844   848   852   864   868   872   874   876
          880   884   888   892   897   901   905   909   911   913   917   921   925   930
          934   938   940   942   946   950   954   958   963   967   971   975   977   979
          983   987   991  1003  1007  1011  1013  1015  1019  1023  1027  1031  1036  1040
         1044  1048  1050  1052  1056  1060  1064  1069  1073  1077  1079  1081  1085  1089
         1093  1097  1102  1106  1110  1114  1116  1118  1122  1126  1130  1142  1146  1150
         1152  1154  1158  1162  1166  1170  1175  1179  1183  1187  1189  1191  1195  1199
         1203  1208  1212  1216  1218  1220  1224  1228  1232  1236  1241  1245  1249  1253
         1255  1257  1261  1265  1269  1281  1285  1289  1291  1293  1297  1301  1305  1309
         1314  1318  1322  1326  1328  1330  1334  1338  1342  1347  1351  1355  1357  1359
         1363  1367  1371  1375  1380  1384  1388  1392  1394  1396  1400  1404  1408  1420
         1424  1428  1430  1432  1436  1440  1444  1448  1453  1457  1461  1465  1467  1469
         1473  1477  1481  1486  1490  1494  1496  1498  1502  1506  1510  1514  1519  1523
         1527  1531  1533  1535  1539  1543  1547  1559  1563  1567  1569  1571  1575  1579
         1583  1587  1592  1596  1600  1604  1606  1608  1612  1616  1620  1625  1629  1633
         1635  1637  1641  1645  1649  1653  1658  1662  1666  1670  1672  1674  1678  1682
         1686  1698  1702  1706  1708  1710  1714  1718  1722  1726  1731  1735  1739  1743
         1745  1747  1751  1755  1759  1764  1768  1772  1774  1776  1780  1784  1788  1792
         1797  1801  1805  1809  1811  1813  1817  1821  1825  1837  1841  1845  1847  1849
         1853  1857  1861  1865  1870  1874  1878  1882  1884  1886  1890  1894  1898  1903
         1907  1911  1913  1915  1919  1923  1927  1931  1936  1940  1944  1948  1950  1952
         1956  1960  1964  1976  1980  1984  1986  1988  1992  1996  2000  2004  2009  2013
         2017  2021  2023  2025  2029  2033  2037  2042  2046  2050  2052  2054  2058  2062
         2066  2070  2075  2079  2083  2087  2089  2091  2095  2099  2103  2115  2119  2123
         2125  2127  2131  2135  2139  2143  2148  2152  2156  2160  2162  2164  2168  2172
         2176  2181  2185  2189  2191  2193  2197  2201  2205  2209  2214  2218  2222  2226
         2228  2230  2234  2238  2242  2254  2258  2262  2264  2266  2270  2274  2278  2282
         2287  2291  2295  2299  2301  2303  2307  2311  2315  2320  2324  2328  2330  2332
         2336  2340  2344  2348  2353  2357  2361  2365  2367  2369  2373  2377  2381  2393
         2397  2401  2403  2405  2409  2413  2417  2421  2426  2430  2434  2438  2440  2442
         2446  2450  2454  2459  2463  2467  2469  2471  2475  2479  2483  2487  2492  2496
         2500  2504  2506  2508  2512  2516  2520
.RPLUS    306   367   372   433   445   506   511   572   584   645   650   711   723   784
          789   850   862   923   928   989  1001  1062  1067  1128  1140  1201  1206  1267
         1279  1340  1345  1406  1418  1479  1484  1545  1557  1618  1623  1684  1696  1757
         1762  1823  1835  1896  1901  1962  1974  2035  2040  2101  2113  2174  2179  2240
         2252  2313  2318  2379  2391  2452  2457  2518
.RRMIN    314   359   380   425   453   498   519   564   592   637   658   703   731   776
          797   842   870   915   936   981  1009  1054  1075  1120  1148  1193  1214  1259
         1287  1332  1353  1398  1426  1471  1492  1537  1565  1610  1631  1676  1704  1749
         1770  1815  1843  1888  1909  1954  1982  2027  2048  2093  2121  2166  2187  2232
         2260  2305  2326  2371  2399  2444  2465  2510
.RSOA     150   159   168   177   186   195   204   213   222   231   240   249   258   267
          276   285   303   308   312   316   320   324   328   332   336   341   345   349
          353   357   361   365   369   372   374   378   380   382   386   389   390   394
          396   398   402   407   411   415   419   423   427   431   435   442   447   451
          455   459   463   467   471   475   480   484   488   492   496   500   504   508
```

```
           511   513   517   519   521   525   527   529   533   535   537   541   546   550
           554   558   562   566   570   574   581   586   590   594   598   602   606   610
           614   619   623   627   631   635   639   643   647   650   652   656   658   660
           664   666   668   672   674   676   680   685   689   693   697   701   705   709
           713   720   725   729   733   737   741   745   749   753   758   762   766   770
           774   778   782   786   789   791   795   797   799   803   805   807   811   813
           815   819   824   828   832   836   840   844   848   852   859   864   868   872
           876   880   884   888   892   897   901   905   909   913   917   921   925   928
           930   934   936   938   942   944   946   950   952   954   958   963   967   971
           975   979   983   987   991   998  1003  1007  1011  1015  1019  1023  1027  1031
          1036  1040  1044  1048  1052  1056  1060  1064  1067  1069  1073  1075  1077  1081
          1083  1085  1089  1091  1093  1097  1102  1106  1110  1114  1118  1122  1126  1130
          1137  1142  1146  1150  1154  1158  1162  1166  1170  1175  1179  1183  1187  1191
          1195  1199  1203  1206  1208  1212  1214  1216  1220  1222  1224  1228  1230  1232
          1236  1241  1245  1249  1253  1257  1261  1265  1269  1276  1281  1285  1289  1293
          1297  1301  1305  1309  1314  1318  1322  1326  1330  1334  1338  1342  1345  1347
          1351  1353  1355  1359  1361  1363  1367  1369  1371  1375  1380  1384  1388  1392
          1396  1400  1404  1408  1415  1420  1424  1428  1432  1436  1440  1444  1448  1453
          1457  1461  1465  1469  1473  1477  1481  1484  1486  1490  1492  1494  1498  1500
          1502  1506  1508  1510  1514  1519  1523  1527  1531  1535  1539  1543  1547  1554
          1559  1563  1567  1571  1575  1579  1583  1587  1592  1596  1600  1604  1608  1612
          1616  1620  1623  1625  1629  1631  1633  1637  1639  1641  1645  1647  1649  1653
          1658  1662  1666  1670  1674  1678  1682  1686  1693  1698  1702  1706  1710  1714
          1718  1722  1726  1731  1735  1739  1743  1747  1751  1755  1759  1762  1764  1768
          1770  1772  1776  1778  1780  1784  1786  1788  1792  1797  1801  1805  1809  1813
          1817  1821  1825  1832  1837  1841  1845  1849  1853  1857  1861  1865  1870  1874
          1878  1882  1886  1890  1894  1898  1901  1903  1907  1909  1911  1915  1917  1919
          1923  1925  1927  1931  1936  1940  1944  1948  1952  1956  1960  1964  1971  1976
          1980  1984  1988  1992  1996  2000  2004  2009  2013  2017  2021  2025  2029  2033
          2037  2040  2042  2046  2048  2050  2054  2056  2058  2062  2064  2066  2070  2075
          2079  2083  2087  2091  2095  2099  2103  2110  2115  2119  2123  2127  2131  2135
          2139  2143  2148  2152  2156  2160  2164  2168  2172  2176  2179  2181  2185  2187
          2189  2193  2195  2197  2201  2203  2205  2209  2214  2218  2222  2226  2230  2234
          2238  2242  2249  2254  2258  2262  2266  2270  2274  2278  2282  2287  2291  2295
          2299  2303  2307  2311  2315  2318  2320  2324  2326  2328  2332  2334  2336  2340
          2342  2344  2348  2353  2357  2361  2365  2369  2373  2377  2381  2388  2393  2397
          2401  2405  2409  2413  2417  2421  2426  2430  2434  2438  2442  2446  2450  2454
          2457  2459  2463  2465  2467  2471  2473  2475  2479  2481  2483  2487  2492  2496
          2500  2504  2508  2512  2516  2520
.RSOB      343   351   359   367   482   490   498   506   621   629   637   645   760   768
           776   784   899   907   915   923  1038  1046  1054  1062  1177  1185  1193  1201
          1316  1324  1332  1340  1455  1463  1471  1479  1594  1602  1610  1618  1733  1741
          1749  1757  1872  1880  1888  1896  2011  2019  2027  2035  2150  2158  2166  2174
          2289  2297  2305  2313  2428  2436  2444  2452
.RSOQ      339   347   355   363   478   486   494   502   617   625   633   641   756   764
           772   780   895   903   911   919  1034  1042  1050  1058  1173  1181  1189  1197
          1312  1320  1328  1336  1451  1459  1467  1475  1590  1598  1606  1614  1729  1737
          1745  1753  1868  1876  1884  1892  2007  2015  2023  2031  2146  2154  2162  2170
          2285  2293  2301  2309  2424  2432  2440  2448
.RSAB      310   318   326   334   449   457   465   473   588   596   604   612   727   735
           743   751   866   874   882   890  1005  1013  1021  1029  1144  1152  1160  1168
          1283  1291  1299  1307  1422  1430  1438  1446  1561  1569  1577  1585  1700  1708
          1716  1724  1839  1847  1855  1863  1978  1986  1994  2002  2117  2125  2133  2141
```

```
EQ 1273
           2256   2264   2272   2280   2395   2403   2411   2419
.RSAQ       306    314    322    330    445    453    461    469    584    592    600    608    723    731    SEQ 1286
            739    747    862    870    878    886   1001   1009   1017   1025   1140   1148   1156   1164
           1279   1287   1295   1303   1418   1426   1434   1442   1557   1565   1573   1581   1696   1704
           1712   1720   1835   1843   1851   1859   1974   1982   1990   1998   2113   2121   2129   2137
           2252   2260   2268   2276   2391   2399   2407   2415
.RSDO       111    113    115    117    119    121    123    125    127    129    131    133    135    137
            139    141    143    409    417    425    433    548    556    564    572    687    695    703
            711    826    834    842    850    965    973    981    989   1104   1112   1120   1128   1243
           1251   1259   1267   1382   1390   1398   1406   1521   1529   1537   1545   1660   1668   1676
           1684   1799   1807   1815   1823   1938   1946   1954   1962   2077   2085   2093   2101   2216
           2224   2232   2240   2355   2363   2371   2379   2494   2502   2510   2518
.RSDA       376    384    392    400    515    523    531    539    654    662    670    678    793    801
            809    817    932    940    948    956   1071   1079   1087   1095   1210   1218   1226   1234
           1349   1357   1365   1373   1488   1496   1504   1512   1627   1635   1643   1651   1766   1774
           1782   1790   1905   1913   1921   1929   2044   2052   2060   2068   2183   2191   2199   2207
           2322   2330   2338   2346   2461   2469   2477   2485
.RSDQ       405    413    421    429    544    552    560    568    683    691    699    707    822    830
            838    846    961    969    977    985   1100   1108   1116   1124   1239   1247   1255   1263
           1378   1386   1394   1402   1517   1525   1533   1541   1656   1664   1672   1680   1795   1803
           1811   1819   1934   1942   1950   1958   2073   2081   2089   2097   2212   2220   2228   2236
           2351   2359   2367   2375   2490   2498   2506   2514
.RSKCN      111    113    115    117    119    121    123    125    127    129    131    133    135    137
            139    141    143    306    310    314    318    322    326    330    334    339    343    347
            351    355    359    363    367    372    376    380    384    388    392    396    400    405
            409    413    417    421    425    429    433    445    449    453    457    461    465    469
            473    478    482    486    490    494    498    502    506    511    515    519    523    527
            531    535    539    544    548    552    556    560    564    568    572    584    588    592
            596    600    604    608    612    617    621    625    629    633    637    641    645    650
            654    658    662    666    670    674    678    683    687    691    695    699    703    707
            711    723    727    731    735    739    743    747    751    756    760    764    768    772
            776    780    784    789    793    797    801    805    809    813    817    822    826    830
            834    838    842    846    850    862    866    870    874    878    882    886    890    895
            899    903    907    911    915    919    923    928    932    936    940    944    948    952
            956    961    965    969    973    977    981    985    989   1001   1005   1009   1013   1017
           1021   1025   1029   1034   1038   1042   1046   1050   1054   1058   1062   1067   1071   1075
           1079   1083   1087   1091   1095   1100   1104   1108   1112   1116   1120   1124   1128   1140
           1144   1148   1152   1156   1160   1164   1168   1173   1177   1181   1185   1189   1193   1197
           1201   1206   1210   1214   1218   1222   1226   1230   1234   1239   1243   1247   1251   1255
           1259   1263   1267   1279   1283   1287   1291   1295   1299   1303   1307   1312   1316   1320
           1324   1328   1332   1336   1340   1345   1349   1353   1357   1361   1365   1369   1373   1378
           1382   1386   1390   1394   1398   1402   1406   1418   1422   1426   1430   1434   1438   1442
           1446   1451   1455   1459   1463   1467   1471   1475   1479   1484   1488   1492   1496   1500
           1504   1508   1512   1517   1521   1525   1529   1533   1537   1541   1545   1557   1561   1565
           1569   1573   1577   1581   1585   1590   1594   1598   1602   1606   1610   1614   1618   1623
           1627   1631   1635   1639   1643   1647   1651   1656   1660   1664   1668   1672   1676   1680
           1684   1696   1700   1704   1708   1712   1716   1720   1724   1729   1733   1737   1741   1745
           1749   1753   1757   1762   1766   1770   1774   1778   1782   1786   1790   1795   1799   1803
           1807   1811   1815   1819   1823   1835   1839   1843   1847   1851   1855   1859   1863   1868
           1872   1876   1880   1884   1888   1892   1896   1901   1905   1909   1913   1917   1921   1925
           1929   1934   1938   1942   1946   1950   1954   1958   1962   1974   1978   1982   1986   1990
           1994   1998   2002   2007   2011   2015   2019   2023   2027   2031   2035   2040   2044   2048
           2052   2056   2060   2064   2068   2073   2077   2081   2085   2089   2093   2097   2101   2113
DFP
DFP
```

```
               2117  2121  2125  2129  2133  2137  2141  [illegible]  2150  2154  2158  2162  2166  2170
               2174  2179  2183  2187  2191  2195  2199  [illegible]  2207  2212  2216  2220  2224  2228
               2232  2236  2240  2252  2256  2260  2264  [illegible]  2272  2276  2280  2285  2289  2293
               2297  2301  2305  2309  2313  2318  2322  [illegible]  2330  2334  2338  2342  2346  2351
               2355  2359  2363  2367  2371  2375  2379  [illegible]  2395  2399  2403  2407  2411  2415
               2419  2424  2428  2432  2436  2440  2444  2448  2452  2457  2461  2465  2469  2473
               2477  2481  2485  2490  2494  2498  2502  2506  2510  2514  2518
.RSMIN          310   363   376   429   449   502   515   568   588   641   654   707   727   780
                793   846   866   919   932   985  1005  1058  1071  1124  1144  1197  1210  1263
               1283  1336  1349  1402  1422  1475  1488  1541  1561  1614  1627  1680  1700  1753
               1766  1819  1839  1892  1905  1958  1978  2031  2044  2097  2117  2170  2183  2236
               2256  2309  2322  2375  2395  2448  2461  2514
.RXNOR          334   339   400   405   473   478   539   544   612   617   678   683   751   756
                817   822   890   895   956   961  1029  1034  1095  1100  1168  1173  1234  1239
               1307  1312  1373  1378  1446  1451  1512  1517  1585  1590  1651  1656  1724  1729
               1790  1795  1863  1868  1929  1934  2002  2007  2068  2073  2141  2146  2207  2212
               2280  2285  2346  2351  2419  2424  2485  2490
.RXOR           330   343   396   409   469   482   535   548   608   621   674   687   747   760
                813   826   886   899   952   965  1025  1038  1091  1104  1164  1177  1230  1243
               1303  1316  1369  1382  1442  1455  1508  1521  1581  1594  1647  1660  1720  1733
               1786  1799  1859  1872  1925  1938  1998  2011  2064  2077  2137  2150  2203  2216
               2276  2289  2342  2355  2415  2428  2481  2494
```

```
CALC   103  105  111  113  115  117  119  121  123  125  127  129  131  133
       135  137  139  141  143  145  150  152  154  159  161  163  168  170    SEQ 1288
       172  177  179  181  186  188  190  195  197  199  204  206  208  213
       215  217  222  224  226  231  233  235  240  242  244  249  251  253
       258  260  262  267  269  271  276  278  280  285  287  289  294  296
       301  303  306  308  310  312  314  316  318  320  322  324  326  328
       330  332  334  336  339  341  343  345  347  349  351  353  355  357
       359  361  363  365  367  369  372  374  376  378  380  382  384  386
       388  390  392  394  396  398  400  402  405  407  409  411  413  415
       417  419  421  423  425  427  429  431  433  435  440  442  445  447
       449  451  453  455  457  459  461  463  465  467  469  471  473  475
       478  480  482  484  486  488  490  492  494  496  498  500  502  504
       506  508  511  513  515  517  519  521  523  525  527  529  531  533
       535  537  539  541  544  546  548  550  552  554  556  558  560  562
       564  566  568  570  572  574  579  581  584  586  588  590  592  594
       596  598  600  602  604  606  608  610  612  614  617  619  621  623
       625  627  629  631  633  635  637  639  641  643  645  647  650  652
       654  656  658  660  662  664  666  668  670  672  674  676  678  680
       683  685  687  689  691  693  695  697  699  701  703  705  707  709
       711  713  718  720  723  725  727  729  731  733  735  737  739  741
       743  745  747  749  751  753  756  758  760  762  764  766  768  770
       772  774  776  778  780  782  784  786  789  791  793  795  797  799
       801  803  805  807  809  811  813  815  817  819  822  824  826  828
       830  832  834  836  838  840  842  844  846  848  850  852  857  859
       862  864  866  868  870  872  874  876  878  880  882  884  886  888
       890  892  895  897  899  901  903  905  907  909  911  913  915  917
       919  921  923  925  928  930  932  934  936  938  940  942  944  946
       948  950  952  954  956  958  961  963  965  967  969  971  973  975
       977  979  981  983  985  987  989  991  996  998 1001 1003 1005 1007
      1009 1011 1013 1015 1017 1019 1021 1023 1025 1027 1029 1031 1034 1036
      1038 1040 1042 1044 1046 1048 1050 1052 1054 1056 1058 1060 1062 1064
      1067 1069 1071 1073 1075 1077 1079 1081 1083 1085 1087 1089 1091 1093
      1095 1097 1100 1102 1104 1106 1108 1110 1112 1114 1116 1118 1120 1122
      1124 1126 1128 1130 1135 1137 1140 1142 1144 1146 1148 1150 1152 1154
      1156 1158 1160 1162 1164 1166 1168 1170 1173 1175 1177 1179 1181 1183
      1185 1187 1189 1191 1193 1195 1197 1199 1201 1203 1206 1208 1210 1212
      1214 1216 1218 1220 1222 1224 1226 1228 1230 1232 1234 1236 1239 1241
      1243 1245 1247 1249 1251 1253 1255 1257 1259 1261 1263 1265 1267 1269
      1274 1276 1279 1281 1283 1285 1287 1289 1291 1293 1295 1297 1299 1301
      1303 1305 1307 1309 1312 1314 1316 1318 1320 1322 1324 1326 1328 1330
      1332 1334 1336 1338 1340 1342 1345 1347 1349 1351 1353 1355 1357 1359
      1361 1363 1365 1367 1369 1371 1373 1375 1378 1380 1382 1384 1386 1388
      1390 1392 1394 1396 1398 1400 1402 1404 1406 1408 1413 1415 1418 1420
      1422 1424 1426 1428 1430 1432 1434 1436 1438 1440 1442 1444 1446 1448
      1451 1453 1455 1457 1459 1461 1463 1465 1467 1469 1471 1473 1475 1477
      1479 1481 1484 1486 1488 1490 1492 1494 1496 1498 1500 1502 1504 1506
      1508 1510 1512 1514 1517 1519 1521 1523 1525 1527 1529 1531 1533 1535
      1537 1539 1541 1543 1545 1547 1552 1554 1557 1559 1561 1563 1565 1567
      1569 1571 1573 1575 1577 1579 1581 1583 1585 1587 1590 1592 1594 1596
      1598 1600 1602 1604 1606 1608 1610 1612 1614 1616 1618 1620 1623 1625
      1627 1629 1631 1633 1635 1637 1639 1641 1643 1645 1647 1649 1651 1653
      1656 1658 1660 1662 1664 1666 1668 1670 1672 1674 1676 1678 1680 1682
      1684 1686 1691 1693 1696 1698 1700 1702 1704 1706 1708 1710 1712 1714
```

```
           1716   1718   1720   1722   1724   1726   1729   1731   1733   1735   1737   1739   1741   1743      SEQ 1289
           1745   1747   1749   1751   1753   1755   1757   1759   1762   1764   1766   1768   1770   1772
           1774   1776   1778   1780   1782   1784   1786   1788   1790   1792   1795   1797   1799   1801
           1803   1805   1807   1809   1811   1813   1815   1817   1819   1821   1823   1825   1830   1832
           1835   1837   1839   1841   1843   1845   1847   1849   1851   1853   1855   1857   1859   1861
           1863   1865   1868   1870   1872   1874   1876   1878   1880   1882   1884   1886   1888   1890
           1892   1894   1896   1898   1901   1903   1905   1907   1909   1911   1913   1915   1917   1919
           1921   1923   1925   1927   1929   1931   1934   1936   1938   1940   1942   1944   1946   1948
           1950   1952   1954   1956   1958   1960   1962   1964   1969   1971   1974   1976   1978   1980
           1982   1984   1986   1988   1990   1992   1994   1996   1998   2000   2002   2004   2007   2009
           2011   2013   2015   2017   2019   2021   2023   2025   2027   2029   2031   2033   2035   2037
           2040   2042   2044   2046   2048   2050   2052   2054   2056   2058   2060   2062   2064   2066
           2068   2070   2073   2075   2077   2079   2081   2083   2085   2087   2089   2091   2093   2095
           2097   2099   2101   2103   2108   2110   2113   2115   2117   2119   2121   2123   2125   2127
           2129   2131   2133   2135   2137   2139   2141   2143   2146   2148   2150   2152   2154   2156
           2158   2160   2162   2164   2166   2168   2170   2172   2174   2176   2179   2181   2183   2185
           2187   2189   2191   2193   2195   2197   2199   2201   2203   2205   2207   2209   2212   2214
           2216   2218   2220   2222   2224   2226   2228   2230   2232   2234   2236   2238   2240   2242
           2247   2249   2252   2254   2256   2258   2260   2262   2264   2266   2268   2270   2272   2274
           2276   2278   2280   2282   2285   2287   2289   2291   2293   2295   2297   2299   2301   2303
           2305   2307   2309   2311   2313   2315   2318   2320   2322   2324   2326   2328   2330   2332
           2334   2336   2338   2340   2342   2344   2346   2348   2351   2353   2355   2357   2359   2361
           2363   2365   2367   2369   2371   2373   2375   2377   2379   2381   2386   2388   2391   2393
           2395   2397   2399   2401   2403   2405   2407   2409   2411   2413   2415   2417   2419   2421
           2424   2426   2428   2430   2432   2434   2436   2438   2440   2442   2444   2446   2448   2450
           2452   2454   2457   2459   2461   2463   2465   2467   2469   2471   2473   2475   2477   2479
           2481   2483   2485   2487   2490   2492   2494   2496   2498   2500   2502   2504   2506   2508
           2510   2512   2514   2516   2518   2520   2522
CONCAT      103    105    111    113    115    117    119    121    123    125    127    129    131    133
            135    137    139    141    143    145    150    152    154    159    161    163    168    170
            172    177    179    181    186    188    190    195    197    199    204    206    208    213
            215    217    222    224    226    231    233    235    240    242    244    249    251    253
            258    260    262    267    269    271    276    278    280    285    287    289    294    296
            301    303    306    308    310    312    314    316    318    320    322    324    326    328
            330    332    334    336    339    341    343    345    347    349    351    353    355    357
            359    361    363    365    367    369    372    374    376    378    380    382    384    386
            388    390    392    394    396    398    400    402    405    407    409    411    413    415
            417    419    421    423    425    427    429    431    433    435    440    442    445    447
            449    451    453    455    457    459    461    463    465    467    469    471    473    475
            478    480    482    484    486    488    490    492    494    496    498    500    502    504
            506    508    511    513    515    517    519    521    523    525    527    529    531    533
            535    537    539    541    544    546    548    550    552    554    556    558    560    562
            564    566    568    570    572    574    579    581    584    586    588    590    592    594
            596    598    600    602    604    606    608    610    612    614    617    619    621    623
            625    627    629    631    633    635    637    639    641    643    645    647    650    652
            654    656    658    660    662    664    666    668    670    672    674    676    678    680
            683    685    687    689    691    693    695    697    699    701    703    705    707    709
            711    713    718    720    723    725    727    729    731    733    735    737    739    741
            743    745    747    749    751    753    756    758    760    762    764    766    768    770
            772    774    776    778    780    782    784    786    789    791    793    795    797    799
            801    803    805    807    809    811    813    815    817    819    822    824    826    828
            830    832    834    836    838    840    842    844    846    848    850    852    857    859
            862    864    866    868    870    872    874    876    878    880    882    884    886    888
```

```
  890   892   895   897   899   901   903   905   907   909   911   913   915   917      SEQ 1290
  919   921   923   925   928   930   932   934   936   938   940   942   944   946
  948   950   952   954   956   958   961   963   965   967   969   971   973   975
  977   979   981   983   985   987   989   991   996   998  1001  1003  1005  1007
 1009  1011  1013  1015  1017  1019  1021  1023  1025  1027  1029  1031  1034  1036
 1038  1040  1042  1044  1046  1048  1050  1052  1054  1056  1058  1060  1062  1064
 1067  1069  1071  1073  1075  1077  1079  1081  1083  1085  1087  1089  1091  1093
 1095  1097  1100  1102  1104  1106  1108  1110  1112  1114  1116  1118  1120  1122
 1124  1126  1128  1130  1135  1137  1140  1142  1144  1146  1148  1150  1152  1154
 1156  1158  1160  1162  1164  1166  1168  1170  1173  1175  1177  1179  1181  1183
 1185  1187  1189  1191  1193  1195  1197  1199  1201  1203  1206  1208  1210  1212
 1214  1216  1218  1220  1222  1224  1226  1228  1230  1232  1234  1236  1239  1241
 1243  1245  1247  1249  1251  1253  1255  1257  1259  1261  1263  1265  1267  1269
 1274  1276  1279  1281  1283  1285  1287  1289  1291  1293  1295  1297  1299  1301
 1303  1305  1307  1309  1312  1314  1316  1318  1320  1322  1324  1326  1328  1330
 1332  1334  1336  1338  1340  1342  1345  1347  1349  1351  1353  1355  1357  1359
 1361  1363  1365  1367  1369  1371  1373  1375  1378  1380  1382  1384  1386  1388
 1390  1392  1394  1396  1398  1400  1402  1404  1406  1408  1413  1415  1418  1420
 1422  1424  1426  1428  1430  1432  1434  1436  1438  1440  1442  1444  1446  1448
 1451  1453  1455  1457  1459  1461  1463  1465  1467  1469  1471  1473  1475  1477
 1479  1481  1484  1486  1488  1490  1492  1494  1496  1498  1500  1502  1504  1506
 1508  1510  1512  1514  1517  1519  1521  1523  1525  1527  1529  1531  1533  1535
 1537  1539  1541  1543  1545  1547  1552  1554  1557  1559  1561  1563  1565  1567
 1569  1571  1573  1575  1577  1579  1581  1583  1585  1587  1590  1592  1594  1596
 1598  1600  1602  1604  1606  1608  1610  1612  1614  1616  1618  1620  1623  1625
 1627  1629  1631  1633  1635  1637  1639  1641  1643  1645  1647  1649  1651  1653
 1656  1658  1660  1662  1664  1666  1668  1670  1672  1674  1676  1678  1680  1682
 1684  1686  1691  1693  1696  1698  1700  1702  1704  1706  1708  1710  1712  1714
 1716  1718  1720  1722  1724  1726  1729  1731  1733  1735  1737  1739  1741  1743
 1745  1747  1749  1751  1753  1755  1757  1759  1762  1764  1766  1768  1770  1772
 1774  1776  1778  1780  1782  1784  1786  1788  1790  1792  1795  1797  1799  1801
 1803  1805  1807  1809  1811  1813  1815  1817  1819  1821  1823  1825  1830  1832
 1835  1837  1839  1841  1843  1845  1847  1849  1851  1853  1855  1857  1859  1861
 1863  1865  1868  1870  1872  1874  1876  1878  1880  1882  1884  1886  1888  1890
 1892  1894  1896  1898  1901  1903  1905  1907  1909  1911  1913  1915  1917  1919
 1921  1923  1925  1927  1929  1931  1934  1936  1938  1940  1942  1944  1946  1948
 1950  1952  1954  1956  1958  1960  1962  1964  1969  1971  1974  1976  1978  1980
 1982  1984  1986  1988  1990  1992  1994  1996  1998  2000  2002  2004  2007  2009
 2011  2013  2015  2017  2019  2021  2023  2025  2027  2029  2031  2033  2035  2037
 2040  2042  2044  2046  2048  2050  2052  2054  2056  2058  2060  2062  2064  2066
 2068  2070  2073  2075  2077  2079  2081  2083  2085  2087  2089  2091  2093  2095
 2097  2099  2101  2103  2108  2110  2113  2115  2117  2119  2121  2123  2125  2127
 2129  2131  2133  2135  2137  2139  2141  2143  2146  2148  2150  2152  2154  2156
 2158  2160  2162  2164  2166  2168  2170  2172  2174  2176  2179  2181  2183  2185
 2187  2189  2191  2193  2195  2197  2199  2201  2203  2205  2207  2209  2212  2214
 2216  2218  2220  2222  2224  2226  2228  2230  2232  2234  2236  2238  2240  2242
 2247  2249  2252  2254  2256  2258  2260  2262  2264  2266  2268  2270  2272  2274
 2276  2278  2280  2282  2285  2287  2289  2291  2293  2295  2297  2299  2301  2303
 2305  2307  2309  2311  2313  2315  2318  2320  2322  2324  2326  2328  2330  2332
 2334  2336  2338  2340  2342  2344  2346  2348  2351  2353  2355  2357  2359  2361
 2363  2365  2367  2369  2371  2373  2375  2377  2379  2381  2386  2388  2391  2393
 2395  2397  2399  2401  2403  2405  2407  2409  2411  2413  2415  2417  2419  2421
 2424  2426  2428  2430  2432  2434  2436  2438  2440  2442  2444  2446  2448  2450
```

```
         2452   2454   2457   2459   2461   2463   2465   2467   2469   2471   2473   2475   2477   2479     SEQ 1291
         2481   2483   2485   2487   2490   2492   2494   2496   2498   2500   2502   2504   2506   2508
         2510   2512   2514   2516   2518   2520   2522
FIELD     103    105    111    113    115    117    119    121    123    125    127    129    131    133
          135    137    139    141    143    145    150    152    154    159    161    163    168    170
          172    177    179    181    186    188    190    195    197    199    204    206    208    213
          215    217    222    224    226    231    233    235    240    242    244    249    251    253
          258    260    262    267    269    271    276    278    280    285    287    289    294    296
          301    303    306    308    310    312    314    316    318    320    322    324    326    328
          330    332    334    336    339    341    343    345    347    349    351    353    355    357
          359    361    363    365    367    369    372    374    376    378    380    382    384    386
          388    390    392    394    396    398    400    402    405    407    409    411    413    415
          417    419    421    423    425    427    429    431    433    435    440    442    445    447
          449    451    453    455    457    459    461    463    465    467    469    471    473    475
          478    480    482    484    486    488    490    492    494    496    498    500    502    504
          506    508    511    513    515    517    519    521    523    525    527    529    531    533
          535    537    539    541    544    546    548    550    552    554    556    558    560    562
          564    566    568    570    572    574    579    581    584    586    588    590    592    594
          596    598    600    602    604    606    608    610    612    614    617    619    621    623
          625    627    629    631    633    635    637    639    641    643    645    647    650    652
          654    656    658    660    662    664    666    668    670    672    674    676    678    680
          683    685    687    689    691    693    695    697    699    701    703    705    707    709
          711    713    718    720    723    725    727    729    731    733    735    737    739    741
          743    745    747    749    751    753    756    758    760    762    764    766    768    770
          772    774    776    778    780    782    784    786    789    791    793    795    797    799
          801    803    805    807    809    811    813    815    817    819    822    824    826    828
          830    832    834    836    838    840    842    844    846    848    850    852    857    859
          862    864    866    868    870    872    874    876    878    880    882    884    886    888
          890    892    895    897    899    901    903    905    907    909    911    913    915    917
          919    921    923    925    928    930    932    934    936    938    940    942    944    946
          948    950    952    954    956    958    961    963    965    967    969    971    973    975
          977    979    981    983    985    987    989    991    996    998   1001   1003   1005   1007
         1009   1011   1013   1015   1017   1019   1021   1023   1025   1027   1029   1031   1034   1036
         1038   1040   1042   1044   1046   1048   1050   1052   1054   1056   1058   1060   1062   1064
         1067   1069   1071   1073   1075   1077   1079   1081   1083   1085   1087   1089   1091   1093
         1095   1097   1100   1102   1104   1106   1108   1110   1112   1114   1116   1118   1120   1122
         1124   1126   1128   1130   1135   1137   1140   1142   1144   1146   1148   1150   1152   1154
         1156   1158   1160   1162   1164   1166   1168   1170   1173   1175   1177   1179   1181   1183
         1185   1187   1189   1191   1193   1195   1197   1199   1201   1203   1206   1208   1210   1212
         1214   1216   1218   1220   1222   1224   1226   1228   1230   1232   1234   1236   1239   1241
         1243   1245   1247   1249   1251   1253   1255   1257   1259   1261   1263   1265   1267   1269
         1274   1276   1279   1281   1283   1285   1287   1289   1291   1293   1295   1297   1299   1301
         1303   1305   1307   1309   1312   1314   1316   1318   1320   1322   1324   1326   1328   1330
         1332   1334   1336   1338   1340   1342   1345   1347   1349   1351   1353   1355   1357   1359
         1361   1363   1365   1367   1369   1371   1373   1375   1378   1380   1382   1384   1386   1388
         1390   1392   1394   1396   1398   1400   1402   1404   1406   1408   1413   1415   1418   1420
         1422   1424   1426   1428   1430   1432   1434   1436   1438   1440   1442   1444   1446   1448
         1451   1453   1455   1457   1459   1461   1463   1465   1467   1469   1471   1473   1475   1477
         1479   1481   1484   1486   1488   1490   1492   1494   1496   1498   1500   1502   1504   1506
         1508   1510   1512   1514   1517   1519   1521   1523   1525   1527   1529   1531   1533   1535
         1537   1539   1541   1543   1545   1547   1552   1554   1557   1559   1561   1563   1565   1567
         1569   1571   1573   1575   1577   1579   1581   1583   1585   1587   1590   1592   1594   1596
         1598   1600   1602   1604   1606   1608   1610   1612   1614   1616   1618   1620   1623   1625
         1627   1629   1631   1633   1635   1637   1639   1641   1643   1645   1647   1649   1651   1653      SEQ 1292
         1656   1658   1660   1662   1664   1666   1668   1670   1672   1674   1676   1678   1680   1682
         1684   1686   1691   1693   1696   1698   1700   1702   1704   1706   1708   1710   1712   1714
         1716   1718   1720   1722   1724   1726   1729   1731   1733   1735   1737   1739   1741   1743
         1745   1747   1749   1751   1753   1755   1757   1759   1762   1764   1766   1768   1770   1772
         1774   1776   1778   1780   1782   1784   1786   1788   1790   1792   1795   1797   1799   1801
         1803   1805   1807   1809   1811   1813   1815   1817   1819   1821   1823   1825   1830   1832
         1835   1837   1839   1841   1843   1845   1847   1849   1851   1853   1855   1857   1859   1861
         1863   1865   1868   1870   1872   1874   1876   1878   1880   1882   1884   1886   1888   1890
         1892   1894   1896   1898   1901   1903   1905   1907   1909   1911   1913   1915   1917   1919
         1921   1923   1925   1927   1929   1931   1934   1936   1938   1940   1942   1944   1946   1948
         1950   1952   1954   1956   1958   1960   1962   1964   1969   1971   1974   1976   1978   1980
         1982   1984   1986   1988   1990   1992   1994   1996   1998   2000   2002   2004   2007   2009
         2011   2013   2015   2017   2019   2021   2023   2025   2027   2029   2031   2033   2035   2037
         2040   2042   2044   2046   2048   2050   2052   2054   2056   2058   2060   2062   2064   2066
         2068   2070   2073   2075   2077   2079   2081   2083   2085   2087   2089   2091   2093   2095
         2097   2099   2101   2103   2108   2110   2113   2115   2117   2119   2121   2123   2125   2127
         2129   2131   2133   2135   2137   2139   2141   2143   2146   2148   2150   2152   2154   2156
         2158   2160   2162   2164   2166   2168   2170   2172   2174   2176   2179   2181   2183   2185
         2187   2189   2191   2193   2195   2197   2199   2201   2203   2205   2207   2209   2212   2214
         2216   2218   2220   2222   2224   2226   2228   2230   2232   2234   2236   2238   2240   2242
         2247   2249   2252   2254   2256   2258   2260   2262   2264   2266   2268   2270   2272   2274
         2276   2278   2280   2282   2285   2287   2289   2291   2293   2295   2297   2299   2301   2303
         2305   2307   2309   2311   2313   2315   2318   2320   2322   2324   2326   2328   2330   2332
         2334   2336   2338   2340   2342   2344   2346   2348   2351   2353   2355   2357   2359   2361
         2363   2365   2367   2369   2371   2373   2375   2377   2379   2381   2386   2388   2391   2393
         2395   2397   2399   2401   2403   2405   2407   2409   2411   2413   2415   2417   2419   2421
         2424   2426   2428   2430   2432   2434   2436   2438   2440   2442   2444   2446   2448   2450
         2452   2454   2457   2459   2461   2463   2465   2467   2469   2471   2473   2475   2477   2479
         2481   2483   2485   2487   2490   2492   2494   2496   2498   2500   2502   2504   2506   2508
         2510   2512   2514   2516   2518   2520   2522
GO         75     77     87
MFLD     103#    103   105#    105   111#    111   113#    113   115#    115   117#    117   119#    119
         121#    121   123#    123   125#    125   127#    127   129#    129   131#    131   133#    133
         135#    135   137#    137   139#    139   141#    141   143#    143   145#    145   150#    150
         152#    152   154#    154   159#    159   161#    161   163#    163   168#    168   170#    170
         172#    172   177#    177   179#    179   181#    181   186#    186   188#    188   190#    190
         195#    195   197#    197   199#    199   204#    204   206#    206   208#    208   213#    213
         215#    215   217#    217   222#    222   224#    224   226#    226   231#    231   233#    233
         235#    235   240#    240   242#    242   244#    244   249#    249   251#    251   253#    253
         258#    258   260#    260   262#    262   267#    267   269#    269   271#    271   276#    276
         278#    278   280#    280   285#    285   287#    287   289#    289   294#    294   296#    296
         301#    301   303#    303   306#    306   308#    308   310#    310   312#    312   314#    314
         316#    316   318#    318   320#    320   322#    322   324#    324   326#    326   328#    328
         330#    330   332#    332   334#    334   336#    336   339#    339   341#    341   343#    343
         345#    345   347#    347   349#    349   351#    351   353#    353   355#    355   357#    357
         359#    359   361#    361   363#    363   365#    365   367#    367   369#    369   372#    372
         374#    374   376#    376   378#    378   380#    380   382#    382   384#    384   386#    386
         388#    388   390#    390   392#    392   394#    394   396#    396   398#    398   400#    400
         402#    402   405#    405   407#    407   409#    409   411#    411   413#    413   415#    415
         417#    417   419#    419   421#    421   423#    423   425#    425   427#    427   429#    429
         431#    431   433#    433   435#    435   440#    440   442#    442   445#    445   447#    447
         449#    449   451#    451   453#    453   455#    455   457#    457   459#    459   461#    461
```

```
 463#   463   465#   465   467#   467   469#   469   471#   471   473#   473   475#   475
 478#   478   480#   480   482#   482   484#   484   486#   486   488#   488   490#   490
 492#   492   494#   494   496#   496   498#   498   500#   500   502#   502   504#   504
 506#   506   508#   508   511#   511   513#   513   515#   515   517#   517   519#   519
 521#   521   523#   523   525#   525   527#   527   529#   529   531#   531   533#   533
 535#   535   537#   537   539#   539   541#   541   544#   544   546#   546   548#   548
 550#   550   552#   552   554#   554   556#   556   558#   558   560#   560   562#   562
 564#   564   566#   566   568#   568   570#   570   572#   572   574#   574   579#   579
 581#   581   584#   584   586#   586   588#   588   590#   590   592#   592   594#   594
 596#   596   598#   598   600#   600   602#   602   604#   604   606#   606   608#   608
 610#   610   612#   612   614#   614   617#   617   619#   619   621#   621   623#   623
 625#   625   627#   627   629#   629   631#   631   633#   633   635#   635   637#   637
 639#   639   641#   641   643#   643   645#   645   647#   647   650#   650   652#   652
 654#   654   656#   656   658#   658   660#   660   662#   662   664#   664   666#   666
 668#   668   670#   670   672#   672   674#   674   676#   676   678#   678   680#   680
 683#   683   685#   685   687#   687   689#   689   691#   691   693#   693   695#   695
 697#   697   699#   699   701#   701   703#   703   705#   705   707#   707   709#   709
 711#   711   713#   713   718#   718   720#   720   723#   723   725#   725   727#   727
 729#   729   731#   731   733#   733   735#   735   737#   737   739#   739   741#   741
 743#   743   745#   745   747#   747   749#   749   751#   751   753#   753   756#   756
 758#   758   760#   760   762#   762   764#   764   766#   766   768#   768   770#   770
 772#   772   774#   774   776#   776   778#   778   780#   780   782#   782   784#   784
 786#   786   789#   789   791#   791   793#   793   795#   795   797#   797   799#   799
 801#   801   803#   803   805#   805   807#   807   809#   809   811#   811   813#   813
 815#   815   817#   817   819#   819   822#   822   824#   824   826#   826   828#   828
 830#   830   832#   832   834#   834   836#   836   838#   838   840#   840   842#   842
 844#   844   846#   846   848#   848   850#   850   852#   852   857#   857   859#   859
 862#   862   864#   864   866#   866   868#   868   870#   870   872#   872   874#   874
 876#   876   878#   878   880#   880   882#   882   884#   884   886#   886   888#   888
 890#   890   892#   892   895#   895   897#   897   899#   899   901#   901   903#   903
 905#   905   907#   907   909#   909   911#   911   913#   913   915#   915   917#   917
 919#   919   921#   921   923#   923   925#   925   928#   928   930#   930   932#   932
 934#   934   936#   936   938#   938   940#   940   942#   942   944#   944   946#   946
 948#   948   950#   950   952#   952   954#   954   956#   956   958#   958   961#   961
 963#   963   965#   965   967#   967   969#   969   971#   971   973#   973   975#   975
 977#   977   979#   979   981#   981   983#   983   985#   985   987#   987   989#   989
 991#   991   996#   996   998#   998  1001#  1001  1003#  1003  1005#  1005  1007#  1007
1009#  1009  1011#  1011  1013#  1013  1015#  1015  1017#  1017  1019#  1019  1021#  1021
1023#  1023  1025#  1025  1027#  1027  1029#  1029  1031#  1031  1034#  1034  1036#  1036
1038#  1038  1040#  1040  1042#  1042  1044#  1044  1046#  1046  1048#  1048  1050#  1050
1052#  1052  1054#  1054  1056#  1056  1058#  1058  1060#  1060  1062#  1062  1064#  1064
1067#  1067  1069#  1069  1071#  1071  1073#  1073  1075#  1075  1077#  1077  1079#  1079
1081#  1081  1083#  1083  1085#  1085  1087#  1087  1089#  1089  1091#  1091  1093#  1093
1095#  1095  1097#  1097  1100#  1100  1102#  1102  1104#  1104  1106#  1106  1108#  1108
1110#  1110  1112#  1112  1114#  1114  1116#  1116  1118#  1118  1120#  1120  1122#  1122
1124#  1124  1126#  1126  1128#  1128  1130#  1130  1135#  1135  1137#  1137  1140#  1140
1142#  1142  1144#  1144  1146#  1146  1148#  1148  1150#  1150  1152#  1152  1154#  1154
1156#  1156  1158#  1158  1160#  1160  1162#  1162  1164#  1164  1166#  1166  1168#  1168
1170#  1170  1173#  1173  1175#  1175  1177#  1177  1179#  1179  1181#  1181  1183#  1183
1185#  1185  1187#  1187  1189#  1189  1191#  1191  1193#  1193  1195#  1195  1197#  1197
1199#  1199  1201#  1201  1203#  1203  1206#  1206  1208#  1208  1210#  1210  1212#  1212
1214#  1214  1216#  1216  1218#  1218  1220#  1220  1222#  1222  1224#  1224  1226#  1226
1228#  1228  1230#  1230  1232#  1232  1234#  1234  1236#  1236  1239#  1239  1241#  1241
```

```
1243#  1243     1245#  1245     1247#  1247     1249#  1249     1251#  1251     1253#  1253     1255#  1255
1257#  1257     1259#  1259     1261#  1261     1263#  1263     1265#  1265     1267#  1267     1269#  1269
1274#  1274     1276#  1276     1279#  1279     1281#  1281     1283#  1283     1285#  1285     1287#  1287
1289#  1289     1291#  1291     1293#  1293     1295#  1295     1297#  1297     1299#  1299     1301#  1301
1303#  1303     1305#  1305     1307#  1307     1309#  1309     1312#  1312     1314#  1314     1316#  1316
1318#  1318     1320#  1320     1322#  1322     1324#  1324     1326#  1326     1328#  1328     1330#  1330
1332#  1332     1334#  1334     1336#  1336     1338#  1338     1340#  1340     1342#  1342     1345#  1345
1347#  1347     1349#  1349     1351#  1351     1353#  1353     1355#  1355     1357#  1357     1359#  1359
1361#  1361     1363#  1363     1365#  1365     1367#  1367     1369#  1369     1371#  1371     1373#  1373
1375#  1375     1378#  1378     1380#  1380     1382#  1382     1384#  1384     1386#  1386     1388#  1388
1390#  1390     1392#  1392     1394#  1394     1396#  1396     1398#  1398     1400#  1400     1402#  1402
1404#  1404     1406#  1406     1408#  1408     1413#  1413     1415#  1415     1418#  1418     1420#  1420
1422#  1422     1424#  1424     1426#  1426     1428#  1428     1430#  1430     1432#  1432     1434#  1434
1436#  1436     1438#  1438     1440#  1440     1442#  1442     1444#  1444     1446#  1446     1448#  1448
1451#  1451     1453#  1453     1455#  1455     1457#  1457     1459#  1459     1461#  1461     1463#  1463
1465#  1465     1467#  1467     1469#  1469     1471#  1471     1473#  1473     1475#  1475     1477#  1477
1479#  1479     1481#  1481     1484#  1484     1486#  1486     1488#  1488     1490#  1490     1492#  1492
1494#  1494     1496#  1496     1498#  1498     1500#  1500     1502#  1502     1504#  1504     1506#  1506
1508#  1508     1510#  1510     1512#  1512     1514#  1514     1517#  1517     1519#  1519     1521#  1521
1523#  1523     1525#  1525     1527#  1527     1529#  1529     1531#  1531     1533#  1533     1535#  1535
1537#  1537     1539#  1539     1541#  1541     1543#  1543     1545#  1545     1547#  1547     1552#  1552
1554#  1554     1557#  1557     1559#  1559     1561#  1561     1563#  1563     1565#  1565     1567#  1567
1569#  1569     1571#  1571     1573#  1573     1575#  1575     1577#  1577     1579#  1579     1581#  1581
1583#  1583     1585#  1585     1587#  1587     1590#  1590     1592#  1592     1594#  1594     1596#  1596
1598#  1598     1600#  1600     1602#  1602     1604#  1604     1606#  1606     1608#  1608     1610#  1610
1612#  1612     1614#  1614     1616#  1616     1618#  1618     1620#  1620     1623#  1623     1625#  1625
1627#  1627     1629#  1629     1631#  1631     1633#  1633     1635#  1635     1637#  1637     1639#  1639
1641#  1641     1643#  1643     1645#  1645     1647#  1647     1649#  1649     1651#  1651     1653#  1653
1656#  1656     1658#  1658     1660#  1660     1662#  1662     1664#  1664     1666#  1666     1668#  1668
1670#  1670     1672#  1672     1674#  1674     1676#  1676     1678#  1678     1680#  1680     1682#  1682
1684#  1684     1686#  1686     1691#  1691     1693#  1693     1696#  1696     1698#  1698     1700#  1700
1702#  1702     1704#  1704     1706#  1706     1708#  1708     1710#  1710     1712#  1712     1714#  1714
1716#  1716     1718#  1718     1720#  1720     1722#  1722     1724#  1724     1726#  1726     1729#  1729
1731#  1731     1733#  1733     1735#  1735     1737#  1737     1739#  1739     1741#  1741     1743#  1743
1745#  1745     1747#  1747     1749#  1749     1751#  1751     1753#  1753     1755#  1755     1757#  1757
1759#  1759     1762#  1762     1764#  1764     1766#  1766     1768#  1768     1770#  1770     1772#  1772
1774#  1774     1776#  1776     1778#  1778     1780#  1780     1782#  1782     1784#  1784     1786#  1786
1788#  1788     1790#  1790     1792#  1792     1795#  1795     1797#  1797     1799#  1799     1801#  1801
1803#  1803     1805#  1805     1807#  1807     1809#  1809     1811#  1811     1813#  1813     1815#  1815
1817#  1817     1819#  1819     1821#  1821     1823#  1823     1825#  1825     1830#  1830     1832#  1832
1835#  1835     1837#  1837     1839#  1839     1841#  1841     1843#  1843     1845#  1845     1847#  1847
1849#  1849     1851#  1851     1853#  1853     1855#  1855     1857#  1857     1859#  1859     1861#  1861
1863#  1863     1865#  1865     1868#  1868     1870#  1870     1872#  1872     1874#  1874     1876#  1876
1878#  1878     1880#  1880     1882#  1882     1884#  1884     1886#  1886     1888#  1888     1890#  1890
1892#  1892     1894#  1894     1896#  1896     1898#  1898     1901#  1901     1903#  1903     1905#  1905
1907#  1907     1909#  1909     1911#  1911     1913#  1913     1915#  1915     1917#  1917     1919#  1919
1921#  1921     1923#  1923     1925#  1925     1927#  1927     1929#  1929     1931#  1931     1934#  1934
1936#  1936     1938#  1938     1940#  1940     1942#  1942     1944#  1944     1946#  1946     1948#  1948
1950#  1950     1952#  1952     1954#  1954     1956#  1956     1958#  1958     1960#  1960     1962#  1962
1964#  1964     1969#  1969     1971#  1971     1974#  1974     1976#  1976     1978#  1978     1980#  1980
1982#  1982     1984#  1984     1986#  1986     1988#  1988     1990#  1990     1992#  1992     1994#  1994
1996#  1996     1998#  1998     2000#  2000     2002#  2002     2004#  2004     2007#  2007     2009#  2009
2011#  2011     2013#  2013     2015#  2015     2017#  2017     2019#  2019     2021#  2021     2023#  2023
```

```
        2025#  2025   2027#  2027   2029#  2029   2031#  2031   2033#  2033   2035#  2035   2037#  2037      SEQ 1295
        2040#  2040   2042#  2042   2044#  2044   2046#  2046   2048#  2048   2050#  2050   2052#  2052
        2054#  2054   2056#  2056   2058#  2058   2060#  2060   2062#  2062   2064#  2064   2066#  2066
        2068#  2068   2070#  2070   2073#  2073   2075#  2075   2077#  2077   2079#  2079   2081#  2081
        2083#  2083   2085#  2085   2087#  2087   2089#  2089   2091#  2091   2093#  2093   2095#  2095
        2097#  2097   2099#  2099   2101#  2101   2103#  2103   2108#  2108   2110#  2110   2113#  2113
        2115#  2115   2117#  2117   2119#  2119   2121#  2121   2123#  2123   2125#  2125   2127#  2127
        2129#  2129   2131#  2131   2133#  2133   2135#  2135   2137#  2137   2139#  2139   2141#  2141
        2143#  2143   2146#  2146   2148#  2148   2150#  2150   2152#  2152   2154#  2154   2156#  2156
        2158#  2158   2160#  2160   2162#  2162   2164#  2164   2166#  2166   2168#  2168   2170#  2170
        2172#  2172   2174#  2174   2176#  2176   2179#  2179   2181#  2181   2183#  2183   2185#  2185
        2187#  2187   2189#  2189   2191#  2191   2193#  2193   2195#  2195   2197#  2197   2199#  2199
        2201#  2201   2203#  2203   2205#  2205   2207#  2207   2209#  2209   2212#  2212   2214#  2214
        2216#  2216   2218#  2218   2220#  2220   2222#  2222   2224#  2224   2226#  2226   2228#  2228
        2230#  2230   2232#  2232   2234#  2234   2236#  2236   2238#  2238   2240#  2240   2242#  2242
        2247#  2247   2249#  2249   2252#  2252   2254#  2254   2256#  2256   2258#  2258   2260#  2260
        2262#  2262   2264#  2264   2266#  2266   2268#  2268   2270#  2270   2272#  2272   2274#  2274
        2276#  2276   2278#  2278   2280#  2280   2282#  2282   2285#  2285   2287#  2287   2289#  2289
        2291#  2291   2293#  2293   2295#  2295   2297#  2297   2299#  2299   2301#  2301   2303#  2303
        2305#  2305   2307#  2307   2309#  2309   2311#  2311   2313#  2313   2315#  2315   2318#  2318
        2320#  2320   2322#  2322   2324#  2324   2326#  2326   2328#  2328   2330#  2330   2332#  2332
        2334#  2334   2336#  2336   2338#  2338   2340#  2340   2342#  2342   2344#  2344   2346#  2346
        2348#  2348   2351#  2351   2353#  2353   2355#  2355   2357#  2357   2359#  2359   2361#  2361
        2363#  2363   2365#  2365   2367#  2367   2369#  2369   2371#  2371   2373#  2373   2375#  2375
        2377#  2377   2379#  2379   2381#  2381   2386#  2386   2388#  2388   2391#  2391   2393#  2393
        2395#  2395   2397#  2397   2399#  2399   2401#  2401   2403#  2403   2405#  2405   2407#  2407
        2409#  2409   2411#  2411   2413#  2413   2415#  2415   2417#  2417   2419#  2419   2421#  2421
        2424#  2424   2426#  2426   2428#  2428   2430#  2430   2432#  2432   2434#  2434   2436#  2436
        2438#  2438   2440#  2440   2442#  2442   2444#  2444   2446#  2446   2448#  2448   2450#  2450
        2452#  2452   2454#  2454   2457#  2457   2459#  2459   2461#  2461   2463#  2463   2465#  2465
        2467#  2467   2469#  2469   2471#  2471   2473#  2473   2475#  2475   2477#  2477   2479#  2479
        2481#  2481   2483#  2483   2485#  2485   2487#  2487   2490#  2490   2492#  2492   2494#  2494
        2496#  2496   2498#  2498   2500#  2500   2502#  2502   2504#  2504   2506#  2506   2508#  2508
        2510#  2510   2512#  2512   2514#  2514   2516#  2516   2518#  2518   2520#  2520   2522#  2522
MWORD    103    105    111    113    115    117    119    121    123    125    127    129    131    133
         135    137    139    141    143    145    150    152    154    159    161    163    168    170
         172    177    179    181    186    188    190    195    197    199    204    206    208    213
         215    217    222    224    226    231    233    235    240    242    244    249    251    253
         258    260    262    267    269    271    276    278    280    285    287    289    294    296
         301    303    306    308    310    312    314    316    318    320    322    324    326    328
         330    332    334    336    339    341    343    345    347    349    351    353    355    357
         359    361    363    365    367    369    372    374    376    378    380    382    384    386
         388    390    392    394    396    398    400    402    405    407    409    411    413    415
         417    419    421    423    425    427    429    431    433    435    440    442    445    447
         449    451    453    455    457    459    461    463    465    467    469    471    473    475
         478    480    482    484    486    488    490    492    494    496    498    500    502    [illegible]
         506    508    511    513    515    517    519    521    523    525    527    529    531    533
         535    537    539    541    544    546    548    550    552    554    556    558    560    562
         564    566    568    570    572    574    579    581    584    586    588    590    592    594
         596    598    600    602    604    606    608    610    612    614    617    619    621    623
         625    627    629    631    633    635    637    639    641    643    645    647    650    652
         654    656    658    660    662    664    666    668    670    672    674    676    678    680
         683    685    687    689    691    693    695    697    699    701    703    705    707    709
```

711	713	718	720	723	725	727	729	731	733	735	737	739	741
743	745	747	749	751	753	756	758	760	762	764	766	768	770
772	774	776	778	780	782	784	786	789	791	793	795	797	799
801	803	805	807	809	811	813	815	817	819	822	824	826	828
830	832	834	836	838	840	842	844	846	848	850	852	857	859
862	864	866	868	870	872	874	876	878	880	882	884	886	888
890	892	895	897	899	901	903	905	907	909	911	913	915	917
919	921	923	925	928	930	932	934	936	938	940	942	944	946
948	950	952	954	956	958	961	963	965	967	969	971	973	975
977	979	981	983	985	987	989	991	996	998	1001	1003	1005	1007
1009	1011	1013	1015	1017	1019	1021	1023	1025	1027	1029	1031	1034	1036
1038	1040	1042	1044	1046	1048	1050	1052	1054	1056	1058	1060	1062	1064
1067	1069	1071	1073	1075	1077	1079	1081	1083	1085	1087	1089	1091	1093
1095	1097	1100	1102	1104	1106	1108	1110	1112	1114	1116	1118	1120	1122
1124	1126	1128	1130	1135	1137	1140	1142	1144	1146	1148	1150	1152	1154
1156	1158	1160	1162	1164	1166	1168	1170	1173	1175	1177	1179	1181	1183
1185	1187	1189	1191	1193	1195	1197	1199	1201	1203	1206	1208	1210	1212
1214	1216	1218	1220	1222	1224	1226	1228	1230	1232	1234	1236	1239	1241
1243	1245	1247	1249	1251	1253	1255	1257	1259	1261	1263	1265	1267	1269
1274	1276	1279	1281	1283	1285	1287	1289	1291	1293	1295	1297	1299	1301
1303	1305	1307	1309	1312	1314	1316	1318	1320	1322	1324	1326	1328	1330
1332	1334	1336	1338	1340	1342	1345	1347	1349	1351	1353	1355	1357	1359
1361	1363	1365	1367	1369	1371	1373	1375	1378	1380	1382	1384	1386	1388
1390	1392	1394	1396	1398	1400	1402	1404	1406	1408	1413	1415	1418	1420
1422	1424	1426	1428	1430	1432	1434	1436	1438	1440	1442	1444	1446	1448
1451	1453	1455	1457	1459	1461	1463	1465	1467	1469	1471	1473	1475	1477
1479	1481	1484	1486	1488	1490	1492	1494	1496	1498	1500	1502	1504	1506
1508	1510	1512	1514	1517	1519	1521	1523	1525	1527	1529	1531	1533	1535
1537	1539	1541	1543	1545	1547	1552	1554	1557	1559	1561	1563	1565	1567
1569	1571	1573	1575	1577	1579	1581	1583	1585	1587	1590	1592	1594	1596
1598	1600	1602	1604	1606	1608	1610	1612	1614	1616	1618	1620	1623	1625
1627	1629	1631	1633	1635	1637	1639	1641	1643	1645	1647	1649	1651	1653
1656	1658	1660	1662	1664	1666	1668	1670	1672	1674	1676	1678	1680	1682
1684	1686	1691	1693	1696	1698	1700	1702	1704	1706	1708	1710	1712	1714
1716	1718	1720	1722	1724	1726	1729	1731	1733	1735	1737	1739	1741	1743
1745	1747	1749	1751	1753	1755	1757	1759	1762	1764	1766	1768	1770	1772
1774	1776	1778	1780	1782	1784	1786	1788	1790	1792	1795	1797	1799	1801
1803	1805	1807	1809	1811	1813	1815	1817	1819	1821	1823	1825	1830	1832
1835	1837	1839	1841	1843	1845	1847	1849	1851	1853	1855	1857	1859	1861
1863	1865	1868	1870	1872	1874	1876	1878	1880	1882	1884	1886	1888	1890
1892	1894	1896	1898	1901	1903	1905	1907	1909	1911	1913	1915	1917	1919
1921	1923	1925	1927	1929	1931	1934	1936	1938	1940	1942	1944	1946	1948
1950	1952	1954	1956	1958	1960	1962	1964	1969	1971	1974	1976	1978	1980
1982	1984	1986	1988	1990	1992	1994	1996	1998	2000	2002	2004	2007	2009
2011	2013	2015	2017	2019	2021	2023	2025	2027	2029	2031	2033	2035	2037
2040	2042	2044	2046	2048	2050	2052	2054	2056	2058	2060	2062	2064	2066
2068	2070	2073	2075	2077	2079	2081	2083	2085	2087	2089	2091	2093	2095
2097	2099	2101	2103	2108	2110	2113	2115	2117	2119	2121	2123	2125	2127
2129	2131	2133	2135	2137	2139	2141	2143	2146	2148	2150	2152	2154	2156
2158	2160	2162	2164	2166	2168	2170	2172	2174	2176	2179	2181	2183	2185
2187	2189	2191	2193	2195	2197	2199	2201	2203	2205	2207	2209	2212	2214
2216	2218	2220	2222	2224	2226	2228	2230	2232	2234	2236	2238	2240	2242
2247	2249	2252	2254	2256	2258	2260	2262	2264	2266	2268	2270	2272	2274

```
             2276   2278   2280   2282   2285   2287   2289   2291   2293   2295   2297   2299   2301   2303
             2305   2307   2309   2311   2313   2315   2318   2320   2322   2324   2326   2328   2330   2332
             2334   2336   2338   2340   2342   2344   2346   2348   2351   2353   2355   2357   2359   2361
             2363   2365   2367   2369   2371   2373   2375   2377   2379   2381   2386   2388   2391   2393
             2395   2397   2399   2401   2403   2405   2407   2409   2411   2413   2415   2417   2419   2421
             2424   2426   2428   2430   2432   2434   2436   2438   2440   2442   2444   2446   2448   2450
             2452   2454   2457   2459   2461   2463   2465   2467   2469   2471   2473   2475   2477   2479
             2481   2483   2485   2487   2490   2492   2494   2496   2498   2500   2502   2504   2506   2508
             2510   2512   2514   2516   2518   2520   2522
RTN            78     95
SCOPER         89
```

```
                                                TITLE   DFPTA7
                                                SUBTTL  EBUS/MPROC 2901 Tests (Part 4)

                                                SALL

                                                SEARCH  DFPTA,MONSYM

                                        ; INTERN's

                                                ENTRY   TSTA33,TSTA34,TSTA35,TSTA36,TSTA37,TSTA40,TSTA41,TSTA42

                                        ; EXTERN's located in DFPTA1.MAC (Mainline code..)

                                                EXTERN  TLOAD,TRACE,TSTSUB,AEXEC,AAPNT,ALEBF,ALFLS,BEXEC,BBPNT

                                        ; EXTERN's located in DFPTAI.MAC (Port Device Handling Routines module)

                                                EXTERN  IPACLR

                                        ; EXTERN's located in DFPTAU.MAC (Utility Routines module)

                                                EXTERN  SCOSW


                                        ;#*********************************************************************
                                        ; Z7 - Address for use in DDT
                                        ;#*********************************************************************

        000000'                         Z7:                                     ; address of 00000'
```

```
                                      ;#**********************************************************************
                                      ;* TEST 33 - Q-Register Shift
                                      ;
                                      ; Description:  Verify that the Q-register can  be  shifted  left
                                      ;               and right correctly.
                                      ;
                                      ; Procedure:    Clear Port
                                      ;               Load microcode
                                      ;
                                      ;               Set RAR to 0
                                      ;               Execute JMAP,CONT,JMAP - load a 1 into the Q-Reg
                                      ;                 and write into EBUF to verify initialization
                                      ;                 worked.
                                      ;               Read EBUF (Contains contents of Reg Q)
                                      ;               Verify data is correct - 0000..001
                                      ;
                                      ;               Set RAR to 3
                                      ;               Execute JMAP,JMAP sequence
                                      ;               Read EBUF (Contains contents of Q-Reg)
                                      ;               Verify data is correct 0000..002
                                      ;
                                      ;               Repeat the above 4 steps 34 times and verify the
                                      ;                 bit is shifting left one position each time.
                                      ;
                                      ;               Set RAR to 5
                                      ;               Execute JMAP - copy 400000,,0 into Q-Reg
                                      ;               Verify data is correct 4000..000
                                      ;
                                      ;               Set RAR to 5
                                      ;               Execute JMAP,JMAP sequence
                                      ;               Read EBUF (Contains contents of Q-Reg)
                                      ;               Verify data is correct 2000..000
                                      ;
                                      ;               Repeat the above 4 steps 34 times and verify the
                                      ;                 bit is shifting right one position each time.
                                      ;
                                      ; Failure:      ---
                                      ;#**********************************************************************

                                      ; Test data

000000' 254 00 0 00 000013'           TSTA33: JRST    TG33                    ; go start test
000001' 420402  000033                        EBUS!ALU!NDMP!ZALU!33           ; test mask
000002' 00010   003412'                       T33M,,[ASCIZ ^Q-Register Shift^]
000003' 0034    000000                        [EXP MLAST!E23],,0
000004' 0000    000135'                       TSTA34                          ; failure test table
000005' 000000  000277'                       TSTA35                          ;   ...
000006' 000000  001333'                       TSTA36
000007' 000000  002016'                       TSTA37
000010' 000000  002244'                       TSTA40
000011' 000000  002467'                       TSTA41
000012' 777777  777777                        -1
```

```

                                        ; Start test

 000013' 201 00 0 00 000000'            TG33:   MOVEI   Z7                      ; get address of module start
 000014' 260 17 0 00 000000*                    GO      TRACE                   ; handle trace output
 000015' 201 01 0 00 000104'                    MOVEI   1,T33M                  ; set up microcode address
 000016' 260 17 0 00 000000*                    GO      TLOAD                   ; load/verify it
 000017' 263 17 0 00 000000                     RTN                             ; failed - exit test

                                        ; Initialization

 000020' 400 15 0 00 000000             TL33:   SETZ    ERFLG,                  ; clear error flag
 000021' 260 17 0 00 000000*                    GO      IPACLR                  ; clear port
 000022' 402 00 0 00 000000*                    SETZM   TSTSUB                  ; initialize subtest number
 000023' 201 06 0 00 000035'                    MOVEI   6,TS33                  ; get sstep table address

                                        ; Loop on single step table entries

 000024' 260 17 0 00 000000*            TA33:   GO      BEXEC                   ; execute table entry
 000025' 254 00 0 00 000034'                    JRST    TX33                    ; end of sstep table
 000026' 254 00 0 00 000024'                    JRST    TA33                    ; keep looping after call
 000027' 474 15 0 00 000000                     SETO    ERFLG,                  ; error - EBUF data incorrect

                                        ; Handle error printouts and scope looping

 000030' 027 00 0 00 000102'                    SCOPER  MA33                    ; print error message
 000031' 254 00 0 00 000020'                    JRST    TL33                    ; loop on error
 000032' 254 00 0 00 000034'                    JRST    TX33                    ; altmode exit
 000033' 322 15 0 00 000024'                    JUMPE   ERFLG,TA33              ; do next sstep table entry

                                        ; End of test

 000034' 263 17 0 00 000000             TX33:   RTN                             ; return

                                        ; Sstep table, with EBUF data, as:  (CMD,#Steps,Start addr,End Addr,EBUF data)

 000035' 100300  000002                 TS33:   ATABLE  (SSSTRT,3,0,2,1)        ; do 2 single steps/check results
 000036' 000000  000001
 000037' 300000  000045'                        ATABLE  (SSCALL,TS33IN)         ; initialize test parameters
 000040' 400000  000052'                TS33L:  ATABLE  (SSCHK,TS33AD)          ; set up single step parameters
 000041' 100300  000000                         ATABLE  (SSSTRT,3,0,0,0)        ; do 3 single steps/check results
 000042' 000000  000000
 000043' 500000  000040'                        ATABLE  (SSJRST,TS33L)          ; do next address
 000044' 000000  000000                         ATABLE  (SSLAST)                ; end of table

                                        ; Special routines

 000045' 201 00 0 00 000001             TS33IN: MOVEI   1                       ; initialize data pattern
 000046' 202 00 0 00 003671'                    MOVEM   TS33PT#
 000047' 201 00 0 00 000003                     MOVEI   3                       ; initialize start address
 000050' 202 00 0 00 003672'                    MOVEM   TS33ST#
 000051' 263 17 0 00 000000                     RTN                             ; return
```

```

000052' 200 01 0 00 003671'     TS33AD: MOVE    1,TS33PT                ; get data pattern
000053' 200 02 0 00 003672'             MOVE    2,TS33ST                ; get start address
000054' 326 01 0 00 000062'             JUMPN   1,TS33A0                ; done? no - continue
000055' 306 02 0 00 000011              CAIN    2,11                    ; already on 2nd set?
000056' 263 17 0 00 000000              RTN                             ; yes - return +1 (done)
000057' 515 01 0 00 400000              HRLZI   1,400000                ; no - get proper data pattern
000060' 201 02 0 00 000006              MOVEI   2,6                     ; set up new address
000061' 254 00 0 00 000070'             JRST    TS33A1                  ; continue

000062' 306 02 0 00 000003      TS33A0: CAIN    2,3                     ; left shifting?
000063' 242 01 0 00 000001              LSH     1,1                     ; yes - shift it
000064' 306 02 0 00 000003              CAIN    2,3                     ; left shifting?
000065' 254 00 0 00 000070'             JRST    TS33A1                  ; yes - continue
000066' 201 02 0 00 000011              MOVEI   2,11                    ; set up correct starting address
000067' 242 01 0 00 777777              LSH     1,-1                    ; yes - shift it

000070' 202 01 0 00 003671'     TS33A1: MOVEM   1,TS33PT#               ; save data pattern
000071' 202 02 0 00 003672'             MOVEM   2,TS33ST#               ; save starting address
000072' 200 03 0 06 000000              MOVE    3,(6)                   ; get table entry
000073' 137 02 0 00 003417'             DPB     2,[POINT 12,3,23]       ; save initial address
000074' 201 02 0 02 000002              MOVEI   2,2(2)                  ; get address+1
000075' 137 02 0 00 003420'             DPB     2,[POINT 12,3,35]       ; save final address
000076' 202 03 0 06 000000              MOVEM   3,(6)                   ; save it
000077' 202 01 0 06 000001              MOVEM   1,1(6)                  ; save data pattern
000100' 350 00 0 17 000000              AOS     (P)                     ; set up RTN+2
000101' 263 17 0 00 000000              RTN                             ; return


                                ; Error messages

000102' 140000  003421'         MA33:   MSG!TXNOT![ASCIZ /Q-Register not shifted correctly/]
000103' 000000000000#                   LAST!CALL!TXALL!BBPNT           ; print sequencer data
```

```

                                          ; Microcode:

000104' 000000  010000                    T33M:   MWORD   <ADDR=0,JMAP,J=1,SOA,AND,D=0>             ; 0
000105' 440000  000040
000106' 000100  000000                            MWORD   <CONT,SOQ,PLUS,CRY,D=0>                   ; 1
000107' 200000  000740
000110' 000200  022004                            MWORD   <JMAP,J=2,SOQ,OR,D=1,OENA,SELE,MGC=4>     ; 2
000111' 231000  005040

000112' 000300  040000                            MWORD   <JMAP,J=4,SOQ,OR,D=2,B=1>                 ; 3
000113' 232000  400040
000114' 000400  000000                            MWORD   <CONT,SOQ,OR,D=6>                         ; 4
000115' 236000  000340
000116' 000500  052004                            MWORD   <JMAP,J=5,SOQ,OR,D=1,OENA,SELE,MGC=4>     ; 5
000117' 231000  005040

000120' 000600  070000                            MWORD   <JMAP,J=7,SOA,OR,A=1,D=0>                 ; 6
000121' 430010  000040
000122' 000700  000000                            MWORD   <CONT,SOA,OR,A=1,D=0>                     ; 7
000123' 430010  000340
000124' 001000  102004                            MWORD   <JMAP,J=10,SOQ,OR,D=1,OENA,SELE,MGC=4>    ; 10
000125' 231000  005040

000126' 001100  120000                            MWORD   <JMAP,J=12,SOQ,OR,D=2,B=1>                ; 11
000127' 232000  400040
000130' 001200  000000                            MWORD   <CONT,SOQ,OR,D=4>                         ; 12
000131' 234000  000340
000132' 001300  132004                            MWORD   <JMAP,J=13,SOQ,OR,D=1,OENA,SELE,MGC=4>    ; 13
000133' 231000  005040
000134' 777777  777777                            -1
```

```

                                        ;#**********************************************************************
                                        ;* TEST 34 - Q-Register Right/Left Shift
                                        ;
                                        ; Description:  Verify that the Q-register can  be  shifted  left
                                        ;               and right correctly at full speed.
                                        ;
                                        ; Procedure:    Clear Port
                                        ;               Load microcode
                                        ;
                                        ;               Set RAR to 0
                                        ;               Execute JMAP,CONT - load a 1 into the Q-Reg
                                        ;
                                        ;               Execute a loop of 35 left shifts followed by
                                        ;                 35 right shifts, with 5 left/right shifts at
                                        ;                 each point, and verify that the final data
                                        ;                 in the Q-Reg, read out of the EBUF is 00..001
                                        ;
                                        ; Failure:      ---
                                        ;#**********************************************************************

                                        ; Test data

000135' 254 00 0 00 000147'             TSTA34: JRST    TG34                    ; go start test
000136' 420402  000034                          EBUS!ALU!NDMP!ZALU!34           ; test mask
000137' 000210' 003430'                         T34M,,[ASCIZ ^Q-Register Right/Left Shift^]
000140' 003416' 000000                          [EXP MLAST!E23],,0
000141' 000000  000277'                         TSTA35                          ; failure test table
000142' 000000  001333'                         TSTA36                          ;   ...
000143' 000000  002016'                         TSTA37
000144' 000000  002244'                         TSTA40
000145' 000000  002467'                         TSTA41
000146' 777777  777777                          -1

                                        ; Start test

000147' 201 00 0 00 000000'             TG34:   MOVEI   Z7                      ; get address of module start
000150' 260 17 0 00 000014*                     GO      TRACE                   ; handle trace output
000151' 201 01 0 00 000210'                     MOVEI   1,T34M                  ; set up microcode address
000152' 260 17 0 00 000016*                     GO      TLOAD                   ; load/verify it
000153' 263 17 0 00 000000                      RTN                             ; failed - exit test

                                        ; Initialization

000154' 400 15 0 00 000000              TL34:   SETZ    ERFLG,                  ; clear error flag
000155' 260 17 0 00 000021*                     GO      IPACLR                  ; clear port
```

```

                                       ; 1st segment of test (Segment A) - Start up port and wait till it stops

000156' 201 01 0 00 000012             TA34:   MOVEI   1,^D10                  ; maximum delay before timeout (msec)
000157' 201 02 0 00 000000                     MOVEI   2,0                     ; error address
000160' 201 03 0 00 000032                     MOVEI   3,32                    ; correct address
000161' 400 04 0 00 000000                     SETZ    4,                      ; starting address
000162' 260 17 0 00 000000*                    GO      AEXEC                   ; execute ALU type test
000163' 255 00 0 00 000000                     JFCL                            ; ignore error
000164' 400 15 0 00 000000                     SETZ    ERFLG,                  ; clear error flag
000165' 332 00 0 00 000000*                    SKIPE   ALFLS                   ; started up ok?
000166' 474 15 0 00 000000                     SETO    ERFLG,                  ; no - set error flag
000167' 200 00 0 00 000000*                    MOVE    ALEBF                   ; get EBUF data
000170' 302 00 0 00 000001                     CAIE    1                       ; correct?
000171' 474 15 0 00 000000                     SETO    ERFLG,                  ; no - set error flag

                                       ; Handle error printouts and scope looping

000172' 027 00 0 00 000176'                    SCOPER  MA34                    ; print error message
000173' 254 00 0 00 000154'                    JRST    TL34                    ; loop on error
000174' 254 00 0 00 000175'                    JRST    TX34                    ; altmode exit

                                       ; End of test

000175' 263 17 0 00 000000             TX34:   RTN                             ; return

                                       ; Error messages

000176' 140000  003436'                MA34:   MSG!TXNOT![ASCIZ /Q-Register not shifted correctly at full speed/]
000177' 270000  000200'                        LAST!CALL!TXALL!MA34PN

000200' 200 01 0 00 000000*            MA34PN: MOVE    1,SCOSW                 ; get switches
000201' 603 01 0 00 000200                     TLNE    1,TXTINH                ; full printout
                                               TMSGC   <EBUF (C):  000000 000001
000202' 037 00 0 00 003450'                (A):  >
000203' 607 01 0 00 000200                     TLNN    1,TXTINH                ; full printout
                                               TMSGC   <EBUF (Correct):  000000 000001
000204' 037 00 0 00 003460'                 (Actual):  >
000205' 200 00 0 00 000167*                    MOVE    ALEBF
000206' 037 13 0 00 000000                     PNTHW
000207' 263 17 0 00 000000                     RTN

                                       ; Microcode:

000210' 000000  010000                 T34M:   MWORD   <ADDR=0,JMAP,J=1,SOA,AND,D=0>           ; 0
000211' 440000  000040
000212' 000100  000000                         MWORD   <CONT,SOQ,PLUS,CRY,D=0>                 ; 1
000213' 200000  000740

                                       ; Left shifts

000214' 000200  420000                         MWORD   <LDCT,J=42,D=1>                         ; 2
000215' 001000  000300
000216' 000300  000000                         MWORD   <CONT,SOQ,OR,D=6>                       ; 3
000217' 236000  000340
```

```
000220' 000400  000000                    MWORD   <CONT,SOQ,OR,D=4>                        ; 4
000221' 234000  000340
000222' 000500  000000                    MWORD   <CONT,SOQ,OR,D=6>                        ; 5
000223' 236000  000340
000224' 000600  000000                    MWORD   <CONT,SOQ,OR,D=4>                        ; 6
000225' 234000  000340
000226' 000700  030000                    MWORD   <RPCT,J=3,SOQ,OR,D=6>                    ; 7
000227' 236000  000220

                                  ; Right shifts

000230' 001000  420000                    MWORD   <LDCT,J=42,D=1>                          ; 10
000231' 001000  000300
000232' 001100  000000                    MWORD   <CONT,SOQ,OR,D=4>                        ; 11
000233' 234000  000340
000234' 001200  000000                    MWORD   <CONT,SOQ,OR,D=6>                        ; 12
000235' 236000  000340
000236' 001300  000000                    MWORD   <CONT,SOQ,OR,D=4>                        ; 13
000237' 234000  000340
000240' 001400  000000                    MWORD   <CONT,SOQ,OR,D=6>                        ; 14
000241' 236000  000340
000242' 001500  110000                    MWORD   <RPCT,J=11,SOQ,OR,D=4>                   ; 15
000243' 234000  000220

                                  ; Left shifts

000244' 001600  420000                    MWORD   <LDCT,J=42,D=1>                          ; 16
000245' 001000  000300
000246' 001700  000000                    MWORD   <CONT,SOQ,OR,D=6>                        ; 17
000247' 236000  000340
000250' 002000  000000                    MWORD   <CONT,SOQ,OR,D=4>                        ; 20
000251' 234000  000340
000252' 002100  000000                    MWORD   <CONT,SOQ,OR,D=6>                        ; 21
000253' 236000  000340
000254' 002200  000000                    MWORD   <CONT,SOQ,OR,D=4>                        ; 22
000255' 234000  000340
000256' 002300  170000                    MWORD   <RPCT,J=17,SOQ,OR,D=6>                   ; 23
000257' 236000  000220

                                  ; Right shifts

000260' 002400  420000                    MWORD   <LDCT,J=42,D=1>                          ; 24
000261' 001000  000300
000262' 002500  000000                    MWORD   <CONT,SOQ,OR,D=4>                        ; 25
000263' 234000  000340
000264' 002600  000000                    MWORD   <CONT,SOQ,OR,D=6>                        ; 26
000265' 236000  000340
000266' 002700  000000                    MWORD   <CONT,SOQ,OR,D=4>                        ; 27
000267' 234000  000340
000270' 003000  000000                    MWORD   <CONT,SOQ,OR,D=6>                        ; 30
000271' 236000  000340
000272' 003100  250000                    MWORD   <RPCT,J=25,SOQ,OR,D=4>                   ; 31
000273' 234000  000220

000274' 003200  322004                    MWORD   <JMAP,J=32,SOQ,OR,D=1,OENA,SELE,MGC=4>   ; 32
```

```
   354  000275' 231000  005040
   355  000276' 777777  777777                  -1
```

```
                                      ;#**********************************************************************
                                      ;* TEST 35 - RAM Register Shift
                                      ;
                                      ; Description:  Verify that the RAM registers can be shifted left
                                      ;               and right correctly.
                                      ;
                                      ; Procedure:    Clear Port
                                      ;               Load microcode
                                      ;
                                      ;               Set RAR to 0
                                      ;               Execute JMAP,CONT,JMAP - load a 1 into Reg 0
                                      ;                 and write into EBUF to verify initialization
                                      ;                 worked.
                                      ;               Read EBUF (Contains contents of Reg 0)
                                      ;               Verify data is correct - 0000..001
                                      ;
                                      ;               Set RAR to 3
                                      ;               Execute JMAP,JMAP sequence
                                      ;               Read EBUF (Contains contents of Reg 0)
                                      ;               Verify data is correct 0000..002
                                      ;
                                      ;               Repeat the above 4 steps 34 times and verify the
                                      ;                 bit is shifting left one position each time.
                                      ;
                                      ;               Set RAR to 5
                                      ;               Execute JMAP - copy 400000,,0 into Reg 0
                                      ;               Verify data is correct 4000..000
                                      ;
                                      ;               Set RAR to 5
                                      ;               Execute JMAP,JMAP sequence
                                      ;               Read EBUF (Contains contents of Reg 0)
                                      ;               Verify data is correct 2000..000
                                      ;
                                      ;               Repeat the above 4 steps 34 times and verify the
                                      ;                 bit is shifting right one position each time.
                                      ;
                                      ;               Repeat entire test for each 2901 register 0-17
                                      ;
                                      ; Failure:      ---
                                      ;#**********************************************************************

                                      ; Test data

 000277' 254 00 0 00 000310'          TSTA35: JRST    TG35                    ; go start test
 000300' 420402  000035                       EBUS!ALU!NDMP!ZALU!35           ; test mask
 000301' 000532' 003473'                      T35M,,[ASCIZ ^RAM Register Shift^]
 000302' 003416' 000000                       [EXP MLAST!E23],,0
 000303' 000000  001333'                      TSTA36                          ; failure test table
 000304' 000000  002016'                      TSTA37                          ;   ...
 000305' 000000  002244'                      TSTA40
 000306' 000000  002467'                      TSTA41
 000307' 777777  777777                       -1
```

```

                                        ; Start test

000310' 201 00 0 00 000000'     TG35:   MOVEI   Z7                      ; get address of module start
000311' 260 17 0 00 000150*             GO      TRACE                   ; handle trace output
000312' 201 01 0 00 000532'             MOVEI   1,T35M                  ; set up microcode address
000313' 260 17 0 00 000152*             GO      TLOAD                   ; load/verify it
000314' 263 17 0 00 000000              RTN                             ; failed - exit test

                                        ; Initialization

000315' 400 15 0 00 000000      TL35:   SETZ    ERFLG,                  ; clear error flag
000316' 260 17 0 00 000155*             GO      IPACLR                  ; clear port
000317' 402 00 0 00 000022*             SETZM   TSTSUB                  ; initialize subtest number
000320' 201 06 0 00 000332'             MOVEI   6,TS35                  ; get sstep table address

                                        ; Loop on single step table entries

000321' 260 17 0 00 000024*     TA35:   GO      BEXEC                   ; execute table entry
000322' 254 00 0 00 000331'             JRST    TX35                    ; end of sstep table
000323' 254 00 0 00 000321'             JRST    TA35                    ; keep looping after call
000324' 474 15 0 00 000000              SETO    ERFLG,                  ; error - EBUF data incorrect

                                        ; Handle error printouts and scope looping

000325' 027 00 0 00 000523'             SCOPER  MA35                    ; print error message
000326' 254 00 0 00 000315'             JRST    TL35                    ; loop on error
000327' 254 00 0 00 000331'             JRST    TX35                    ; altmode exit
000330' 322 15 0 00 000321'             JUMPE   ERFLG,TA35              ; do next sstep table entry

                                        ; End of test

000331' 263 17 0 00 000000      TX35:   RTN                             ; return

                                        ; Sstep table, with EBUF data, as:  (CMD,#Steps,Start addr,End Addr,EBUF data)

000332' 300000  000340'         TS35:   ATABLE  (SSCALL,TS35IN)         ; initialize test parameters
000333' 400000  000345'         TS35L:  ATABLE  (SSCHK,TS35AD)          ; set up single step parameters
000334' 100300  000000                  ATABLE  (SSSTRT,3,0,0,0)        ; do 3 single steps/check results
000335' 000000  000000
000336' 500000  000333'                 ATABLE  (SSJRST,TS35L)          ; do next address
000337' 000000  000000                  ATABLE  (SSLAST)                ; end of table

                                        ; Special routines

000340' 476 00 0 00 003674'     TS35IN: SETOM   TS35RG#                 ; initialize register number
000341' 476 00 0 00 003675'             SETOM   TS35ST#                 ; initialize start address pointer
000342' 201 00 0 00 000114              MOVEI   ^D76                    ; initialize pattern number pointer
000343' 202 00 0 00 003673'             MOVEM   TS35PA#
000344' 263 17 0 00 000000              RTN                             ; return
```

```

000345' 200 03 0 00 003675'   TS35AD: MOVE    3,TS35ST                ; get start address pointer
000346' 350 02 0 00 003673'           AOS     2,TS35PA                ; point to next pattern
000347' 200 01 0 02 000405'           MOVE    1,TS35T2(2)             ; get data pattern
000350' 312 01 0 00 003477'           CAME    1,[-2]                  ; done with table?
000351' 254 00 0 00 000360'           JRST    TS35A0                  ; no - proceed
000352' 350 01 0 00 003674'           AOS     1,TS35RG                ; yes - next register
000353' 303 01 0 00 000017            CAILE   1,17                    ; done with test?
000354' 263 17 0 00 000000            RTN                             ; yes - return
000355' 403 03 0 00 003675'           SETZB   3,TS35ST                ; no - initialize start address pointer
000356' 402 00 0 00 003673'           SETZM   TS35PA                  ; initialize pattern number pointer
000357' 254 00 0 00 000364'           JRST    TS35A1

000360' 312 01 0 00 003500'   TS35A0: CAME    1,[-1]                  ; ready for next starting address?
000361' 254 00 0 00 000364'           JRST    TS35A1                  ; no - continue
000362' 350 03 0 00 003675'           AOS     3,TS35ST                ; yes - increment address pointer
000363' 350 00 0 00 003673'           AOS     TS35PA                  ; increment data pattern pointer
000364' 200 01 0 06 000000    TS35A1: MOVE    1,(6)                   ; get table entry
000365' 200 02 0 00 003674'           MOVE    2,TS35RG                ; get register number
000366' 242 02 0 00 000006            LSH     2,6                     ; multiply by 100
000367' 270 02 0 03 000401'           ADD     2,TS35T1(3)             ; get starting address
000370' 137 02 0 00 003501'           DPB     2,[POINT 12,1,23]       ; save initial address
000371' 201 02 0 02 000002            MOVEI   2,2(2)                  ; get address+1
000372' 137 02 0 00 003502'           DPB     2,[POINT 12,1,35]       ; save final address
000373' 202 01 0 06 000000            MOVEM   1,(6)                   ; save it
000374' 200 02 0 00 003673'           MOVE    2,TS35PA                ; get data pattern
000375' 200 01 0 02 000405'           MOVE    1,TS35T2(2)             ; get data pattern
000376' 202 01 0 06 000001            MOVEM   1,1(6)                  ; save data pattern
000377' 350 00 0 17 000000            AOS     (P)                     ; set up RTN+2
000400' 263 17 0 00 000000            RTN                             ; return

                              ; Starting address table

000401' 000000  000000        TS35T1: 0
000402' 000000  000003                3
000403' 000000  000006                6
000404' 000000  000011                11
```

```

                                        ; Data pattern table

 000405' 000000  000001          TS35T2: 1
 000406' 777777  777777                  -1
 000407' 000000  000002                  2
 000410' 000000  000004                  4
 000411' 000000  000010                  10
 000412' 000000  000020                  20
 000413' 000000  000040                  40
 000414' 000000  000100                  100
 000415' 000000  000200                  200
 000416' 000000  000400                  400
 000417' 000000  001000                  1000
 000420' 000000  002000                  2000
 000421' 000000  004000                  4000
 000422' 000000  010000                  10000
 000423' 000000  020000                  20000
 000424' 000000  040000                  40000
 000425' 000000  100000                  100000
 000426' 000000  200000                  200000
 000427' 000000  400000                  400000
 000430' 000001  000000                  1,,0
 000431' 000002  000000                  2,,0
 000432' 000004  000000                  4,,0
 000433' 000010  000000                  10,,0
 000434' 000020  000000                  20,,0
 000435' 000040  000000                  40,,0
 000436' 000100  000000                  100,,0
 000437' 000200  000000                  200,,0
 000440' 000400  000000                  400,,0
 000441' 001000  000000                  1000,,0
 000442' 002000  000000                  2000,,0
 000443' 004000  000000                  4000,,0
 000444' 010000  000000                  10000,,0
 000445' 020000  000000                  20000,,0
 000446' 040000  000000                  40000,,0
 000447' 100000  000000                  100000,,0
 000450' 200000  000000                  200000,,0
 000451' 400000  000000                  400000,,0
 000452' 000000  000000                  0
 000453' 777777  777777                  -1
 000454' 400000  000000                  400000,,0
 000455' 777777  777777                  -1
 000456' 200000  000000                  200000,,0
 000457' 100000  000000                  100000,,0
 000460' 040000  000000                  40000,,0
 000461' 020000  000000                  20000,,0
 000462' 010000  000000                  10000,,0
 000463' 004000  000000                  4000,,0
 000464' 002000  000000                  2000,,0
 000465' 001000  000000                  1000,,0
```

```
000466' 000400  000000                    400,,0
000467' 000200  000000                    200,,0
000470' 000100  000000                    100,,0
000471' 000040  000000                    40,,0
000472' 000020  000000                    20,,0
000473' 000010  000000                    10,,0
000474' 000004  000000                    4,,0
000475' 000002  000000                    2,,0
000476' 000001  000000                    1,,0
000477' 000000  400000                    400000
000500' 000000  200000                    200000
000501' 000000  100000                    100000
000502' 000000  040000                    40000
000503' 000000  020000                    20000
000504' 000000  010000                    10000
000505' 000000  004000                    4000
000506' 000000  002000                    2000
000507' 000000  001000                    1000
000510' 000000  000400                    400
000511' 000000  000200                    200
000512' 000000  000100                    100
000513' 000000  000040                    40
000514' 000000  000020                    20
000515' 000000  000010                    10
000516' 000000  000004                    4
000517' 000000  000002                    2
000520' 000000  000001                    1
000521' 000000  000000                    0
000522' 777777  777776                    -2

                                  ; Error messages

000523' 240000  000525'           MA35:   CALL!TXNOT!MA35PN
000524' 000000000000#                     LAST!CALL!TXALL!BBPNT         ; print sequencer data

000525' 037 00 0 00 003503'       MA35PN: TMSGC   <2901 Register >
000526' 200 00 0 00 003674'               MOVE    TS35RG
000527' 037 16 0 00 000003                PNTOCS
000530' 037 00 0 00 003423'               TMSG    < not shifted correctly>
000531' 263 17 0 00 000000                RTN
```

```

                                        ; Microcode:

                                        ;       Register 0

000532' 000000  010000                  T35M:   MWORD   <ADDR=0,JMAP,J=1,SOB,B=0,AND,D=2>                      ; 0
000533' 342000  000040
000534' 000100  000000                          MWORD   <CONT,SOB,B=0,PLUS,CRY,D=2>                            ; 1
000535' 302000  000740
000536' 000200  022004                          MWORD   <JMAP,J=2,SOA,A=0,OR,D=1,OENA,SELE,MGC=4>              ; 2
000537' 431000  005040
000540' 000300  040000                          MWORD   <JMAP,J=4,SOB,B=0,OR,D=0>                              ; 3
000541' 330000  000040
000542' 000400  000000                          MWORD   <CONT,SOB,B=0,OR,D=7>                                  ; 4
000543' 337000  000340
000544' 000500  052004                          MWORD   <JMAP,J=5,SOA,OR,A=0,D=1,OENA,SELE,MGC=4>              ; 5
000545' 431000  005040
000546' 000600  070000                          MWORD   <JMAP,J=7,SOQ,B=0,OR,D=2>                              ; 6
000547' 232000  000040
000550' 000700  000000                          MWORD   <CONT,SOQ,B=0,OR,D=2>                                  ; 7
000551' 232000  000340
000552' 001000  102004                          MWORD   <JMAP,J=10,SOA,OR,A=0,D=1,OENA,SELE,MGC=4>             ; 10
000553' 431000  005040
000554' 001100  120000                          MWORD   <JMAP,J=12,SOB,B=0,OR,D=0>                             ; 11
000555' 330000  000040
000556' 001200  000000                          MWORD   <CONT,SOB,B=0,OR,D=5>                                  ; 12
000557' 335000  000340
000560' 001300  132004                          MWORD   <JMAP,J=13,SOA,A=0,OR,D=1,OENA,SELE,MGC=4>             ; 13
000561' 431000  005040

                                        ;       Register 1

000562' 010001  010000                          MWORD   <ADDR=100,JMAP,J=101,SOB,B=1,AND,D=2>                  ; 100
000563' 342000  400040
000564' 010100  000000                          MWORD   <CONT,SOB,B=1,PLUS,CRY,D=2>                            ; 101
000565' 302000  400740
000566' 010201  022004                          MWORD   <JMAP,J=102,SOA,A=1,OR,D=1,OENA,SELE,MGC=4>            ; 102
000567' 431010  005040
000570' 010301  040000                          MWORD   <JMAP,J=104,SOB,B=1,OR,D=0>                            ; 103
000571' 330000  400040
000572' 010400  000000                          MWORD   <CONT,SOB,B=1,OR,D=7>                                  ; 104
000573' 337000  400340
000574' 010501  052004                          MWORD   <JMAP,J=105,SOA,OR,A=1,D=1,OENA,SELE,MGC=4>            ; 105
000575' 431010  005040
000576' 010601  070000                          MWORD   <JMAP,J=107,SOQ,B=1,OR,D=2>                            ; 106
000577' 232000  400040
000600' 010700  000000                          MWORD   <CONT,SOQ,B=1,OR,D=2>                                  ; 107
000601' 232000  400340
000602' 011001  102004                          MWORD   <JMAP,J=110,SOA,OR,A=1,D=1,OENA,SELE,MGC=4>            ; 110
000603' 431010  005040
000604' 011101  120000                          MWORD   <JMAP,J=112,SOB,B=1,OR,D=0>                            ; 111
000605' 330000  400040
000606' 011200  000000                          MWORD   <CONT,SOB,B=1,OR,D=5>                                  ; 112
000607' 335000  400340
000610' 011301  132004                          MWORD   <JMAP,J=113,SOA,A=1,OR,D=1,OENA,SELE,MGC=4>            ; 113
```

```
000611' 431010  005040

                                ;       Register 2

000612' 020002  010000                  MWORD   <ADDR=200,JMAP,J=201,SOB,B=2,AND,D=2>                   ; 200
000613' 342001  000040
000614' 020100  000000                  MWORD   <CONT,SOB,B=2,PLUS,CRY,D=2>                             ; 201
000615' 302001  000740
000616' 020202  022004                  MWORD   <JMAP,J=202,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>             ; 202
000617' 431020  005040
000620' 020302  040000                  MWORD   <JMAP,J=204,SOB,B=2,OR,D=0>                             ; 203
000621' 330001  000040
000622' 020400  000000                  MWORD   <CONT,SOB,B=2,OR,D=7>                                   ; 204
000623' 337001  000340
000624' 020502  052004                  MWORD   <JMAP,J=205,SOA,OR,A=2,D=1,OENA,SELE,MGC=4>             ; 205
000625' 431020  005040
000626' 020602  070000                  MWORD   <JMAP,J=207,SOQ,B=2,OR,D=2>                             ; 206
000627' 232001  000040
000630' 020700  000000                  MWORD   <CONT,SOQ,B=2,OR,D=2>                                   ; 207
000631' 232001  000340
000632' 021002  102004                  MWORD   <JMAP,J=210,SOA,OR,A=2,D=1,OENA,SELE,MGC=4>             ; 210
000633' 431020  005040
000634' 021102  120000                  MWORD   <JMAP,J=212,SOB,B=2,OR.D=0>                             ; 211
000635' 330001  000040
000636' 021200  000000                  MWORD   <CONT,SOB,B=2,OR,D=5>                                   ; 212
000637' 335001  000340
000640' 021302  132004                  MWORD   <JMAP,J=213,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>             ; 213
000641' 431020  005040

                                ;       Register 3

000642' 030003  010000                  MWORD   <ADDR=300,JMAP,J=301,SOB,B=3,AND,D=2>                   ; 300
000643' 342001  400040
000644' 030100  000000                  MWORD   <CONT,SOB,B=3,PLUS,CRY,D=2>                             ; 301
000645' 302001  400740
000646' 030203  022004                  MWORD   <JMAP,J=302,SOA,A=3,OR,D=1,OENA,SELE,MGC=4>             ; 302
000647' 431030  005040
000650' 030303  040000                  MWORD   <JMAP,J=304,SOB,B=3,OR,D=0>                             ; 303
000651' 330001  400040
000652' 030400  000000                  MWORD   <CONT,SOB,B=3,OR,D=7>                                   ; 304
000653' 337001  400340
000654' 030503  052004                  MWORD   <JMAP,J=305,SOA,OR,A=3,D=1,OENA,SELE,MGC=4>             ; 305
000655' 431030  005040
000656' 030603  070000                  MWORD   <JMAP,J=307,SOQ,B=3,OR,D=2>                             ; 306
000657' 232001  400040
000660' 030700  000000                  MWORD   <CONT,SOQ,B=3,OR,D=2>                                   ; 307
000661' 232001  400340
000662' 031003  102004                  MWORD   <JMAP,J=310,SOA,OR,A=3,D=1,OENA,SELE,MGC=4>             ; 310
000663' 431030  005040
000664' 031103  120000                  MWORD   <JMAP,J=312,SOB,B=3,OR,D=0>                             ; 311
000665' 330001  400040
000666' 031200  000000                  MWORD   <CONT,SOB,B=3,OR,D=5>                                   ; 312
000667' 335001  400340
000670' 031303  132004                  MWORD   <JMAP,J=313,SOA,A=3,OR,D=1,OENA,SELE,MGC=4>             ; 313
000671' 431030  005040
```

```
 698                                                 
 699                                 ;       Register 4
 700
 701  000672' 040004  010000                 MWORD   <ADDR=400,JMAP,J=401,SOB,B=4,AND,D=2>               ; 400
 702  000673' 342002  000040
 703  000674' 040100  000000                 MWORD   <CONT,SOB,B=4,PLUS,CRY,D=2>                         ; 401
 704  000675' 302002  000740
 705  000676' 040204  022004                 MWORD   <JMAP,J=402,SOA,A=4,OR,D=1,OENA,SELE,MGC=4>         ; 402
 706  000677' 431040  005040
 707  000700' 040304  040000                 MWORD   <JMAP,J=404,SOB,B=4,OR,D=0>                         ; 403
 708  000701' 330002  000040
 709  000702' 040400  000000                 MWORD   <CONT,SOB,B=4,OR,D=7>                               ; 404
 710  000703' 337002  000340
 711  000704' 040504  052004                 MWORD   <JMAP,J=405,SOA,OR,A=4,D=1,OENA,SELE,MGC=4>         ; 405
 712  000705' 431040  005040
 713  000706' 040604  070000                 MWORD   <JMAP,J=407,SOQ,B=4,OR,D=2>                         ; 406
 714  000707' 232002  000040
 715  000710' 040700  000000                 MWORD   <CONT,SOQ,B=4,OR,D=2>                               ; 407
 716  000711' 232002  000340
 717  000712' 041004  102004                 MWORD   <JMAP,J=410,SOA,OR,A=4,D=1,OENA,SELE,MGC=4>         ; 410
 718  000713' 431040  005040
 719  000714' 041104  120000                 MWORD   <JMAP,J=412,SOB,B=4,OR,D=0>                         ; 411
 720  000715' 330002  000040
 721  000716' 041200  000000                 MWORD   <CONT,SOB,B=4,OR,D=5>                               ; 412
 722  000717' 335002  000340
 723  000720' 041304  132004                 MWORD   <JMAP,J=413,SOA,A=4,OR,D=1,OENA,SELE,MGC=4>         ; 413
 724  000721' 431040  005040
 725
 726                                 ;       Register 5
 727
 728  000722' 050005  010000                 MWORD   <ADDR=500,JMAP,J=501,SOB,B=5,AND,D=2>               ; 500
 729  000723' 342002  400040
 730  000724' 050100  000000                 MWORD   <CONT,SOB,B=5,PLUS,CRY,D=2>                         ; 501
 731  000725' 302002  400740
 732  000726' 050205  022004                 MWORD   <JMAP,J=502,SOA,A=5,OR,D=1,OENA,SELE,MGC=4>         ; 502
 733  000727' 431050  005040
 734  000730' 050305  040000                 MWORD   <JMAP,J=504,SOB,B=5,OR,D=0>                         ; 503
 735  000731' 330002  400040
 736  000732' 050400  000000                 MWORD   <CONT,SOB,B=5,OR,D=7>                               ; 504
 737  000733' 337002  400340
 738  000734' 050505  052004                 MWORD   <JMAP,J=505,SOA,OR,A=5,D=1,OENA,SELE,MGC=4>         ; 505
 739  000735' 431050  005040
 740  000736' 050605  070000                 MWORD   <JMAP,J=507,SOQ,B=5,OR,D=2>                         ; 506
 741  000737' 232002  400040
 742  000740' 050700  000000                 MWORD   <CONT,SOQ,B=5,OR,D=2>                               ; 507
 743  000741' 232002  400340
 744  000742' 051005  102004                 MWORD   <JMAP,J=510,SOA,OR,A=5,D=1,OENA,SELE,MGC=4>         ; 510
 745  000743' 431050  005040
 746  000744' 051105  120000                 MWORD   <JMAP,J=512,SOB,B=5,OR,D=0>                         ; 511
 747  000745' 330002  400040
 748  000746' 051200  000000                 MWORD   <CONT,SOB,B=5,OR,D=5>                               ; 512
 749  000747' 335002  400340
 750  000750' 051305  132004                 MWORD   <JMAP,J=513,SOA,A=5,OR,D=1,OENA,SELE,MGC=4>         ; 513
 751  000751' 431050  005040
 752
```

```
                                   ;       Register 6

000752' 060006  010000                     MWORD   <ADDR=600,JMAP,J=601,SOB,B=6,AND,D=2>              ; 600
000753' 342003  000040
000754' 060100  000000                     MWORD   <CONT,SOB,B=6,PLUS,CRY,D=2>                        ; 601
000755' 302003  000740
000756' 060206  022004                     MWORD   <JMAP,J=602,SOA,A=6,OR,D=1,OENA,SELE,MGC=4>        ; 602
000757' 431060  005040
000760' 060306  040000                     MWORD   <JMAP,J=604,SOB,B=6,OR,D=0>                        ; 603
000761' 330003  000040
000762' 060400  000000                     MWORD   <CONT,SOB,B=6,OR,D=7>                              ; 604
000763' 337003  000340
000764' 060506  052004                     MWORD   <JMAP,J=605,SOA,OR,A=6,D=1,OENA,SELE,MGC=4>        ; 605
000765' 431060  005040
000766' 060606  070000                     MWORD   <JMAP,J=607,SOQ,B=6,OR,D=2>                        ; 606
000767' 232003  000040
000770' 060700  000000                     MWORD   <CONT,SOQ,B=6,OR,D=2>                              ; 607
000771' 232003  000340
000772' 061006  102004                     MWORD   <JMAP,J=610,SOA,OR,A=6,D=1,OENA,SELE,MGC=4>        ; 610
000773' 431060  005040
000774' 061106  120000                     MWORD   <JMAP,J=612,SOB,B=6,OR,D=0>                        ; 611
000775' 330003  000040
000776' 061200  000000                     MWORD   <CONT,SOB,B=6,OR,D=5>                              ; 612
000777' 335003  000340
001000' 061306  132004                     MWORD   <JMAP,J=613,SOA,A=6,OR,D=1,OENA,SELE,MGC=4>        ; 613
001001' 431060  005040

                                   ;       Register 7

001002' 070007  010000                     MWORD   <ADDR=700,JMAP,J=701,SOB,B=7,AND,D=2>              ; 700
001003' 342003  400040
001004' 070100  000000                     MWORD   <CONT,SOB,B=7,PLUS,CRY,D=2>                        ; 701
001005' 302003  400740
001006' 070207  022004                     MWORD   <JMAP,J=702,SOA,A=7,OR,D=1,OENA,SELE,MGC=4>        ; 702
001007' 431070  005040
001010' 070307  040000                     MWORD   <JMAP,J=704,SOB,B=7,OR,D=0>                        ; 703
001011' 330003  400040
001012' 070400  000000                     MWORD   <CONT,SOB,B=7,OR,D=7>                              ; 704
001013' 337003  400340
001014' 070507  052004                     MWORD   <JMAP,J=705,SOA,OR,A=7,D=1,OENA,SELE,MGC=4>        ; 705
001015' 431070  005040
001016' 070607  070000                     MWORD   <JMAP,J=707,SOQ,B=7,OR,D=2>                        ; 706
001017' 232003  400040
001020' 070700  000000                     MWORD   <CONT,SOQ,B=7,OR,D=2>                              ; 707
001021' 232003  400340
001022' 071007  102004                     MWORD   <JMAP,J=710,SOA,OR,A=7,D=1,OENA,SELE,MGC=4>        ; 710
001023' 431070  005040
001024' 071107  120000                     MWORD   <JMAP,J=712,SOB,B=7,OR,D=0>                        ; 711
001025' 330003  400040
001026' 071200  000000                     MWORD   <CONT,SOB,B=7,OR,D=5>                              ; 712
001027' 335003  400340
001030' 071307  132004                     MWORD   <JMAP,J=713,SOA,A=7,OR,D=1,OENA,SELE,MGC=4>        ; 713
001031' 431070  005040

                                   ;       Register 10
```

```

001032' 100010  010000                  MWORD   <ADDR=1000,JMAP,J=1001,SOB,B=10,AND,D=2>            ; 1000
001033' 342004  000040
001034' 100100  000000                  MWORD   <CONT,SOB,B=10,PLUS,CRY,D=2>                        ; 1001
001035' 302004  000740
001036' 100210  022004                  MWORD   <JMAP,J=1002,SOA,A=10,OR,D=1,OENA,SELE,MGC=4>       ; 1002
001037' 431100  005040
001040' 100310  040000                  MWORD   <JMAP,J=1004,SOB,B=10,OR,D=0>                       ; 1003
001041' 330004  000040
001042' 100400  000000                  MWORD   <CONT,SOB,B=10,OR,D=7>                              ; 1004
001043' 337004  000340
001044' 100510  052004                  MWORD   <JMAP,J=1005,SOA,OR,A=10,D=1,OENA,SELE,MGC=4>       ; 1005
001045' 431100  005040
001046' 100610  070000                  MWORD   <JMAP,J=1007,SOQ,B=10,OR,D=2>                       ; 1006
001047' 232004  000040
001050' 100700  000000                  MWORD   <CONT,SOQ,B=10,OR,D=2>                              ; 1007
001051' 232004  000340
001052' 101010  102004                  MWORD   <JMAP,J=1010,SOA,OR,A=10,D=1,OENA,SELE,MGC=4>       ; 1010
001053' 431100  005040
001054' 101110  120000                  MWORD   <JMAP,J=1012,SOB,B=10,OR,D=0>                       ; 1011
001055' 330004  000040
001056' 101200  000000                  MWORD   <CONT,SOB,B=10,OR,D=5>                              ; 1012
001057' 335004  000340
001060' 101310  132004                  MWORD   <JMAP,J=1013,SOA,A=10,OR,D=1,OENA,SELE,MGC=4>       ; 1013
001061' 431100  005040

                                ;       Register 11

001062' 110011  010000                  MWORD   <ADDR=1100,JMAP,J=1101,SOB,B=11,AND,D=2>            ; 1100
001063' 342004  400040
001064' 110100  000000                  MWORD   <CONT,SOB,B=11,PLUS,CRY,D=2>                        ; 1101
001065' 302004  400740
001066' 110211  022004                  MWORD   <JMAP,J=1102,SOA,A=11,OR,D=1,OENA,SELE,MGC=4>       ; 1102
001067' 431110  005040
001070' 110311  040000                  MWORD   <JMAP,J=1104,SOB,B=11,OR,D=0>                       ; 1103
001071' 330004  400040
001072' 110400  000000                  MWORD   <CONT,SOB,B=11,OR,D=7>                              ; 1104
001073' 337004  400340
001074' 110511  052004                  MWORD   <JMAP,J=1105,SOA,OR,A=11,D=1,OENA,SELE,MGC=4>       ; 1105
001075' 431110  005040
001076' 110611  070000                  MWORD   <JMAP,J=1107,SOQ,B=11,OR,D=2>                       ; 1106
001077' 232004  400040
001100' 110700  000000                  MWORD   <CONT,SOQ,B=11,OR,D=2>                              ; 1107
001101' 232004  400340
001102' 111011  102004                  MWORD   <JMAP,J=1110,SOA,OR,A=11,D=1,OENA,SELE,MGC=4>       ; 1110
001103' 431110  005040
001104' 111111  120000                  MWORD   <JMAP,J=1112,SOB,B=11,OR,D=0>                       ; 1111
001105' 330004  400040
001106' 111200  000000                  MWORD   <CONT,SOB,B=11,OR,D=5>                              ; 1112
001107' 335004  400340
001110' 111311  132004                  MWORD   <JMAP,J=1113,SOA,A=11,OR,D=1,OENA,SELE,MGC=4>       ; 1113
001111' 431110  005040

                                ;       Register 12

```

```
001112' 120012  010000                  MWORD   <ADDR=1200,JMAP,J=1201,SOB,B=12,AND,D=2>        ; 1200
001113' 342005  000040
001114' 120100  000000                  MWORD   <CONT,SOB,B=12,PLUS,CRY,D=2>                    ; 1201
001115' 302005  000740
001116' 120212  022004                  MWORD   <JMAP,J=1202,SOA,A=12,OR,D=1,OENA,SELE,MGC=4>   ; 1202
001117' 431120  005040
001120' 120312  040000                  MWORD   <JMAP,J=1204,SOB,B=12,OR,D=0>                   ; 1203
001121' 330005  000040
001122' 120400  000000                  MWORD   <CONT,SOB,B=12,OR,D=7>                          ; 1204
001123' 337005  000340
001124' 120512  052004                  MWORD   <JMAP,J=1205,SOA,OR,A=12,D=1,OENA,SELE,MGC=4>   ; 1205
001125' 431120  005040
001126' 120612  070000                  MWORD   <JMAP,J=1207,SOQ,B=12,OR,D=2>                   ; 1206
001127' 232005  000040
001130' 120700  000000                  MWORD   <CONT,SOQ,B=12,OR,D=2>                          ; 1207
001131' 232005  000340
001132' 121012  102004                  MWORD   <JMAP,J=1210,SOA,OR,A=12,D=1,OENA,SELE,MGC=4>   ; 1210
001133' 431120  005040
001134' 121112  120000                  MWORD   <JMAP,J=1212,SOB,B=12,OR,D=0>                   ; 1211
001135' 330005  000040
001136' 121200  000000                  MWORD   <CONT,SOB,B=12,OR,D=5>                          ; 1212
001137' 335005  000340
001140' 121312  132004                  MWORD   <JMAP,J=1213,SOA,A=12,OR,D=1,OENA,SELE,MGC=4>   ; 1213
001141' 431120  005040

                                ;       Register 13

001142' 130013  010000                  MWORD   <ADDR=1300,JMAP,J=1301,SOB,B=13,AND,D=2>        ; 1300
001143' 342005  400040
001144' 130100  000000                  MWORD   <CONT,SOB,B=13,PLUS,CRY,D=2>                    ; 1301
001145' 302005  400740
001146' 130213  022004                  MWORD   <JMAP,J=1302,SOA,A=13,OR,D=1,OENA,SELE,MGC=4>   ; 1302
001147' 431130  005040
001150' 130313  040000                  MWORD   <JMAP,J=1304,SOB,B=13,OR,D=0>                   ; 1303
001151' 330005  400040
001152' 130400  000000                  MWORD   <CONT,SOB,B=13,OR,D=7>                          ; 1304
001153' 337005  400340
001154' 130513  052004                  MWORD   <JMAP,J=1305,SOA,OR,A=13,D=1,OENA,SELE,MGC=4>   ; 1305
001155' 431130  005040
001156' 130613  070000                  MWORD   <JMAP,J=1307,SOQ,B=13,OR,D=2>                   ; 1306
001157' 232005  400040
001160' 130700  000000                  MWORD   <CONT,SOQ,B=13,OR,D=2>                          ; 1307
001161' 232005  400340
001162' 131013  102004                  MWORD   <JMAP,J=1310,SOA,OR,A=13,D=1,OENA,SELE,MGC=4>   ; 1310
001163' 431130  005040
001164' 131113  120000                  MWORD   <JMAP,J=1312,SOB,B=13,OR,D=0>                   ; 1311
001165' 330005  400040
001166' 131200  000000                  MWORD   <CONT,SOB,B=13,OR,D=5>                          ; 1312
001167' 335005  400340
001170' 131313  132004                  MWORD   <JMAP,J=1313,SOA,A=13,OR,D=1,OENA,SELE,MGC=4>   ; 1313
001171' 431130  005040

                                ;       Register 14

001172' 140014  010000                  MWORD   <ADDR=1400,JMAP,J=1401,SOB,B=14,AND,D=2>        ; 1400
```

```
 001173' 342006  000040
 001174' 140100  000000                  MWORD   <CONT,SOB,B=14,PLUS,CRY,D=2>                          ; 1401
 001175' 302006  000740
 001176' 140214  022004                  MWORD   <JMAP,J=1402,SOA,A=14,OR,D=1,OENA,SELE,MGC=4>         ; 1402
 001177' 431140  005040
 001200' 140314  040000                  MWORD   <JMAP,J=1404,SOB,B=14,OR,D=0>                         ; 1403
 001201' 330006  000040
 001202' 140400  000000                  MWORD   <CONT,SOB,B=14,OR,D=7>                                ; 1404
 001203' 337006  000340
 001204' 140514  052004                  MWORD   <JMAP,J=1405,SOA,OR,A=14,D=1,OENA,SELE,MGC=4>         ; 1405
 001205' 431140  005040
 001206' 140614  070000                  MWORD   <JMAP,J=1407,SOQ,B=14,OR,D=2>                         ; 1406
 001207' 232006  000040
 001210' 140700  000000                  MWORD   <CONT,SOQ,B=14,OR,D=2>                                ; 1407
 001211' 232006  000340
 001212' 141014  102004                  MWORD   <JMAP,J=1410,SOA,OR,A=14,D=1,OENA,SELE,MGC=4>         ; 1410
 001213' 431140  005040
 001214' 141114  120000                  MWORD   <JMAP,J=1412,SOB,B=14,OR,D=0>                         ; 1411
 001215' 330006  000040
 001216' 141200  000000                  MWORD   <CONT,SOB,B=14,OR,D=5>                                ; 1412
 001217' 335006  000340
 001220' 141314  132004                  MWORD   <JMAP,J=1413,SOA,A=14,OR,D=1,OENA,SELE,MGC=4>         ; 1413
 001221' 431140  005040

                                 ;       Register 15

 001222' 150015  010000                  MWORD   <ADDR=1500,JMAP,J=1501,SOB,B=15,AND,D=2>              ; 1500
 001223' 342006  400040
 001224' 150100  000000                  MWORD   <CONT,SOB,B=15,PLUS,CRY,D=2>                          ; 1501
 001225' 302006  400740
 001226' 150215  022004                  MWORD   <JMAP,J=1502,SOA,A=15,OR,D=1,OENA,SELE,MGC=4>         ; 1502
 001227' 431150  005040
 001230' 150315  040000                  MWORD   <JMAP,J=1504,SOB,B=15,OR,D=0>                         ; 1503
 001231' 330006  400040
 001232' 150400  000000                  MWORD   <CONT,SOB,B=15,OR,D=7>                                ; 1504
 001233' 337006  400340
 001234' 150515  052004                  MWORD   <JMAP,J=1505,SOA,OR,A=15,D=1,OENA,SELE,MGC=4>         ; 1505
 001235' 431150  005040
 001236' 150615  070000                  MWORD   <JMAP,J=1507,SOQ,B=15,OR,D=2>                         ; 1506
 001237' 232006  400040
 001240' 150700  000000                  MWORD   <CONT,SOQ,B=15,OR,D=2>                                ; 1507
 001241' 232006  400340
 001242' 151015  102004                  MWORD   <JMAP,J=1510,SOA,OR,A=15,D=1,OENA,SELE,MGC=4>         ; 1510
 001243' 431150  005040
 001244' 151115  120000                  MWORD   <JMAP,J=1512,SOB,B=15,OR,D=0>                         ; 1511
 001245' 330006  400040
 001246' 151200  000000                  MWORD   <CONT,SOB,B=15,OR,D=5>                                ; 1512
 001247' 335006  400340
 001250' 151315  132004                  MWORD   <JMAP,J=1513,SOA,A=15,OR,D=1,OENA,SELE,MGC=4>         ; 1513
 001251' 431150  005040

                                 ;       Register 16

 001252' 160016  010000                  MWORD   <ADDR=1600,JMAP,J=1601,SOB,B=16,AND,D=2>              ; 1600
 001253' 342007  000040
```

```
001254' 160100  000000          MWORD   <CONT,SOB,B=16,PLUS,CRY,D=2>                    ; 1601
001255' 302007  000740
001256' 160216  022004          MWORD   <JMAP,J=1602,SOA,A=16,OR,D=1,OENA,SELE,MGC=4>   ; 1602
001257' 431160  005040
001260' 160316  040000          MWORD   <JMAP,J=1604,SOB,B=16,OR,D=0>                   ; 1603
001261' 330007  000040
001262' 160400  000000          MWORD   <CONT,SOB,B=16,OR,D=7>                          ; 1604
001263' 337007  000340
001264' 160516  052004          MWORD   <JMAP,J=1605,SOA,OR,A=16,D=1,OENA,SELE,MGC=4>   ; 1605
001265' 431160  005040
001266' 160616  070000          MWORD   <JMAP,J=1607,SOQ,B=16,OR,D=2>                   ; 1606
001267' 232007  000040
001270' 160700  000000          MWORD   <CONT,SOQ,B=16,OR,D=2>                          ; 1607
001271' 232007  000340
001272' 161016  102004          MWORD   <JMAP,J=1610,SOA,OR,A=16,D=1,OENA,SELE,MGC=4>   ; 1610
001273' 431160  005040
001274' 161116  120000          MWORD   <JMAP,J=1612,SOB,B=16,OR,D=0>                   ; 1611
001275' 330007  000040
001276' 161200  000000          MWORD   <CONT,SOB,B=16,OR,D=5>                          ; 1612
001277' 335007  000340
001300' 161316  132004          MWORD   <JMAP,J=1613,SOA,A=16,OR,D=1,OENA,SELE,MGC=4>   ; 1613
001301' 431160  005040

                        ;       Register 17

001302' 170017  010000          MWORD   <ADDR=1700,JMAP,J=1701,SOB,B=17,AND,D=2>        ; 1700
001303' 342007  400040
001304' 170100  000000          MWORD   <CONT,SOB,B=17,PLUS,CRY,D=2>                    ; 1701
001305' 302007  400740
001306' 170217  022004          MWORD   <JMAP,J=1702,SOA,A=17,OR,D=1,OENA,SELE,MGC=4>   ; 1702
001307' 431170  005040
001310' 170317  040000          MWORD   <JMAP,J=1704,SOB,B=17,OR,D=0>                   ; 1703
001311' 330007  400040
001312' 170400  000000          MWORD   <CONT,SOB,B=17,OR,D=7>                          ; 1704
001313' 337007  400340
001314' 170517  052004          MWORD   <JMAP,J=1705,SOA,OR,A=17,D=1,OENA,SELE,MGC=4>   ; 1705
001315' 431170  005040
001316' 170617  070000          MWORD   <JMAP,J=1707,SOQ,B=17,OR,D=2>                   ; 1706
001317' 232007  400040
001320' 170700  000000          MWORD   <CONT,SOQ,B=17,OR,D=2>                          ; 1707
001321' 232007  400340
001322' 171017  102004          MWORD   <JMAP,J=1710,SOA,OR,A=17,D=1,OENA,SELE,MGC=4>   ; 1710
001323' 431170  005040
001324' 171117  120000          MWORD   <JMAP,J=1712,SOB,B=17,OR,D=0>                   ; 1711
001325' 330007  400040
001326' 171200  000000          MWORD   <CONT,SOB,B=17,OR,D=5>                          ; 1712
001327' 335007  400340
001330' 171317  132004          MWORD   <JMAP,J=1713,SOA,A=17,OR,D=1,OENA,SELE,MGC=4>   ; 1713
001331' 431170  005040
001332' 777777  777777          -1
```

```

                                        ;#*****************************************************************
                                        ;* TEST 36 - RAM Data Movement Test
                                        ;
                                        ; Description:    Verify that data can be successfully  moved  from
                                        ;                 RAM location to RAM location at full speed.
                                        ;
                                        ; Procedure:      Clear Port
                                        ;                 Load microcode
                                        ;
                                        ;                 Set RAR to 0
                                        ;                 Execute JMAP - load data pattern into Reg 0
                                        ;                 Execute a sequence of instructions that moves
                                        ;                   the data from register to register while
                                        ;                   clearing other registers.
                                        ;
                                        ;                 When done, read the EBUF and verify that the
                                        ;                   data is correct (516000,,001234).
                                        ;
                                        ; Failure:        ---
                                        ;#*****************************************************************

                                        ; Test data

 001333' 254 00 0 00 001343'            TSTA36: JRST    TG36                    ; go start test
 001334' 420402  000036                         EBUS!ALU!NDMP!ZALU!36           ; test mask
 001335' 001407' 003507'                        T36M,,[ASCIZ ^RAM Data Movement Test^]
 001336' 003416' 000000                         [EXP MLAST!E23],,0
 001337' 000000  002016'                        TSTA37                          ; failure test table
 001340' 000000  002244'                        TSTA40                          ;   ...
 001341' 000000  002467'                        TSTA41
 001342' 777777  777777                         -1

                                        ; Start test

 001343' 201 00 0 00 000000'            TG36:   MOVEI   Z7                      ; get address of module start
 001344' 260 17 0 00 000311*                    GO      TRACE                   ; handle trace output
 001345' 201 01 0 00 001407'                    MOVEI   1,T36M                  ; set up microcode address
 001346' 260 17 0 00 000313*                    GO      TLOAD                   ; load/verify it
 001347' 263 17 0 00 000000                     RTN                             ; failed - exit test

                                        ; Initialization

 001350' 400 15 0 00 000000             TL36:   SETZ    ERFLG,                  ; clear error flag
 001351' 260 17 0 00 000316*                    GO      IPACLR                  ; clear port
```

```

                                ; 1st segment of test (Segment A) - Start up port and wait till it stops

 001352' 201 01 0 00 000012     TA36:   MOVEI   1,^D10                  ; maximum delay before timeout (msec)
 001353' 201 02 0 00 000000             MOVEI   2,0                     ; error address
 001354' 201 03 0 00 000032             MOVEI   3,32                    ; correct address
 001355' 400 04 0 00 000000             SETZ    4,                      ; starting address
 001356' 260 17 0 00 000162*            GO      AEXEC                   ; execute ALU type test
 001357' 255 00 0 00 000000             JFCL                            ; ignore error
 001360' 400 15 0 00 000000             SETZ    ERFLG,                  ; clear error flag
 001361' 332 00 0 00 000165*            SKIPE   ALFLS                   ; started up ok?
 001362' 474 15 0 00 000000             SETO    ERFLG,                  ; no - set error flag
 001363' 200 00 0 00 000205*            MOVE    ALEBF                   ; get EBUF data
 001364' 404 00 0 00 003514'            AND     [777400,,1777]          ; mask out indeterminate bits
 001365' 312 00 0 00 003515'            CAME    [516000,,1234]          ; correct?
 001366' 474 15 0 00 000000             SETO    ERFLG,                  ; no - set error flag

                                ; Handle error printouts and scope looping

 001367' 027 00 0 00 001373'            SCOPER  MA36                    ; print error message
 001370' 254 00 0 00 001350'            JRST    TL36                    ; loop on error
 001371' 254 00 0 00 001372'            JRST    TX36                    ; altmode exit

                                ; End of test

 001372' 263 17 0 00 000000     TX36:   RTN                             ; return

                                ; Error messages

 001373' 140000  003516'        MA36:   MSG!TXNOT![ASCIZ /2901 RAM data got destroyed somehow (result in EBUF)/]
 001374' 270000  001375'                LAST!CALL!TXALL!MA36PN

 001375' 200 01 0 00 000200*    MA36PN: MOVE    1,SCOSW                 ; get switches
 001376' 603 01 0 00 000200             TLNE    1,TXTINH                ; full printout?
                                        TMSGC   <EBUF (C):  000000 000001
 001377' 037 00 0 00 003450'          (A):  >
 001400' 607 01 0 00 000200             TLNN    1,TXTINH                ; full printout?
                                        TMSGC   <EBUF (Correct):  000000 000001
 001401' 037 00 0 00 003460'           (Actual):  >
 001402' 200 00 0 00 001363*            MOVE    ALEBF
 001403' 037 13 0 00 000000             PNTHW
 001404' 607 01 0 00 000200             TLNN    1,TXTINH                ; full printout?
 001405' 037 00 0 00 003531'            TMSGC   <(Middle 16 bits indeterminate)>
 001406' 263 17 0 00 000000             RTN
```

```

                                        ; Microcode:

001407' 000000  011234                  T36M:   MWORD   <ADDR=0,JMAP,J=1,SDO,OR,D=2,SKCN,MGC=1234> ; 0
001410' 732000  240040
001411' 000100  000000                          MWORD   <CONT,SOA,AND,D=2,B=1>                  ; 1
001412' 442000  400340
001413' 000200  000000                          MWORD   <CONT,SOA,PLUS,D=2,A=0,B=1>             ; 2
001414' 402000  400340
001415' 000300  000000                          MWORD   <CONT,SOA,AND,D=2,B=2>                  ; 3
001416' 442001  000340
001417' 000400  000000                          MWORD   <CONT,SOA,PLUS,D=2,A=1,B=2>             ; 4
001420' 402011  000340
001421' 000500  000000                          MWORD   <CONT,SOA,AND,D=2,B=3>                  ; 5
001422' 442001  400340
001423' 000600  000000                          MWORD   <CONT,SOA,PLUS,D=2,A=2,B=3>             ; 6
001424' 402021  400340
001425' 000700  000000                          MWORD   <CONT,SOA,AND,D=2,B=4>                  ; 7
001426' 442002  000340
001427' 001000  000000                          MWORD   <CONT,SOA,PLUS,D=2,A=3,B=4>             ; 10
001430' 402032  000340
001431' 001100  000000                          MWORD   <CONT,SOA,AND,D=2,B=5>                  ; 11
001432' 442002  400340
001433' 001200  000000                          MWORD   <CONT,SOA,PLUS,D=2,A=4,B=5>             ; 12
001434' 402042  400340
001435' 001300  000000                          MWORD   <CONT,SOA,AND,D=2,B=6>                  ; 13
001436' 442003  000340
001437' 001400  000000                          MWORD   <CONT,SOA,PLUS,D=2,A=5,B=6>             ; 14
001440' 402053  000340
001441' 001500  000000                          MWORD   <CONT,SOA,AND,D=2,B=7>                  ; 15
001442' 442003  400340
001443' 001600  000000                          MWORD   <CONT,SOA,PLUS,D=2,A=6,B=7>             ; 16
001444' 402063  400340
001445' 001700  000000                          MWORD   <CONT,SOA,AND,D=2,B=10>                 ; 17
001446' 442004  000340
001447' 002000  000000                          MWORD   <CONT,SOA,PLUS,D=2,A=7,B=10>            ; 20
001450' 402074  000340
001451' 002100  000000                          MWORD   <CONT,SOA,AND,D=2,B=11>                 ; 21
001452' 442004  400340
001453' 002200  000000                          MWORD   <CONT,SOA,PLUS,D=2,A=10,B=11>           ; 22
001454' 402104  400340
001455' 002300  000000                          MWORD   <CONT,SOA,AND,D=2,B=12>                 ; 23
001456' 442005  000340
001457' 002400  000000                          MWORD   <CONT,SOA,PLUS,D=2,A=11,B=12>           ; 24
001460' 402115  000340
001461' 002500  000000                          MWORD   <CONT,SOA,AND,D=2,B=13>                 ; 25
001462' 442005  400340
001463' 002600  000000                          MWORD   <CONT,SOA,PLUS,D=2,A=12,B=13>           ; 26
001464' 402125  400340
001465' 002700  000000                          MWORD   <CONT,SOA,AND,D=2,B=14>                 ; 27
001466' 442006  000340
001467' 003000  000000                          MWORD   <CONT,SOA,PLUS,D=2,A=13,B=14>           ; 30
001470' 402136  000340
001471' 003100  000000                          MWORD   <CONT,SOA,AND,D=2,B=15>                 ; 31
001472' 442006  400340
```

```
 001473' 003200  000000          MWORD   <CONT,SOA,PLUS,D=2,A=14,B=15>           ; 32
 001474' 402146  400340
 001475' 003300  000000          MWORD   <CONT,SOA,AND,D=2,B=16>                 ; 33
 001476' 442007  000340
 001477' 003400  000000          MWORD   <CONT,SOA,PLUS,D=2,A=15,B=16>           ; 34
 001500' 402157  000340
 001501' 003500  000000          MWORD   <CONT,SOA,AND,D=2,B=17>                 ; 35
 001502' 442007  400340
 001503' 003600  000000          MWORD   <CONT,SOA,PLUS,D=2,A=16,B=17>           ; 36
 001504' 402167  400340
 001505' 003700  000000          MWORD   <CONT,SOA,AND,D=2,B=0>                  ; 37
 001506' 442000  000340
 001507' 004000  000000          MWORD   <CONT,SOA,PLUS,D=2,A=17,B=0>            ; 40
 001510' 402170  000340

 001511' 004100  000000          MWORD   <CONT,SOA,AND,D=2,B=1>                  ; 41
 001512' 442000  400340
 001513' 004200  000000          MWORD   <CONT,SOA,OR,D=2,A=0,B=1>               ; 42
 001514' 432000  400340
 001515' 004300  000000          MWORD   <CONT,SOA,AND,D=2,B=2>                  ; 43
 001516' 442001  000340
 001517' 004400  000000          MWORD   <CONT,SOA,OR,D=2,A=1,B=2>               ; 44
 001520' 432011  000340
 001521' 004500  000000          MWORD   <CONT,SOA,AND,D=2,B=3>                  ; 45
 001522' 442001  400340
 001523' 004600  000000          MWORD   <CONT,SOA,OR,D=2,A=2,B=3>               ; 46
 001524' 432021  400340
 001525' 004700  000000          MWORD   <CONT,SOA,AND,D=2,B=4>                  ; 47
 001526' 442002  000340
 001527' 005000  000000          MWORD   <CONT,SOA,OR,D=2,A=3,B=4>               ; 50
 001530' 432032  000340
 001531' 005100  000000          MWORD   <CONT,SOA,AND,D=2,B=5>                  ; 51
 001532' 442002  400340
 001533' 005200  000000          MWORD   <CONT,SOA,OR,D=2,A=4,B=5>               ; 52
 001534' 432042  400340
 001535' 005300  000000          MWORD   <CONT,SOA,AND,D=2,B=6>                  ; 53
 001536' 442003  000340
 001537' 005400  000000          MWORD   <CONT,SOA,OR,D=2,A=5,B=6>               ; 54
 001540' 432053  000340
 001541' 005500  000000          MWORD   <CONT,SOA,AND,D=2,B=7>                  ; 55
 001542' 442003  400340
 001543' 005600  000000          MWORD   <CONT,SOA,OR,D=2,A=6,B=7>               ; 56
 001544' 432063  400340
 001545' 005700  000000          MWORD   <CONT,SOA,AND,D=2,B=10>                 ; 57
 001546' 442004  000340
 001547' 006000  000000          MWORD   <CONT,SOA,OR,D=2,A=7,B=10>              ; 60
 001550' 432074  000340
 001551' 006100  000000          MWORD   <CONT,SOA,AND,D=2,B=11>                 ; 61
 001552' 442004  400340
 001553' 006200  000000          MWORD   <CONT,SOA,OR,D=2,A=10,B=11>             ; 62
 001554' 432104  400340
 001555' 006300  000000          MWORD   <CONT,SOA,AND,D=2,B=12>                 ; 63
 001556' 442005  000340
 001557' 006400  000000          MWORD   <CONT,SOA,OR,D=2,A=11,B=12>             ; 64
 001560' 432115  000340
```

```
   1222 001561' 006500  000000             MWORD  <CONT,SOA,AND,D=2,B=13>            ; 65
   1223 001562' 442005  400340
   1224 001563' 006600  000000             MWORD  <CONT,SOA,OR,D=2,A=12,B=13>        ; 66
   1225 001564' 432125  400340
   1226 001565' 006700  000000             MWORD  <CONT,SOA,AND,D=2,B=14>            ; 67
   1227 001566' 442006  000340
   1228 001567' 007000  000000             MWORD  <CONT,SOA,OR,D=2,A=13,B=14>        ; 70
   1229 001570' 432136  000340
   1230 001571' 007100  000000             MWORD  <CONT,SOA,AND,D=2,B=15>            ; 71
   1231 001572' 442006  400340
   1232 001573' 007200  000000             MWORD  <CONT,SOA,OR,D=2,A=14,B=15>        ; 72
   1233 001574' 432146  400340
   1234 001575' 007300  000000             MWORD  <CONT,SOA,AND,D=2,B=16>            ; 73
   1235 001576' 442007  000340
   1236 001577' 007400  000000             MWORD  <CONT,SOA,OR,D=2,A=15,B=16>        ; 74
   1237 001600' 432157  000340
   1238 001601' 007500  000000             MWORD  <CONT,SOA,AND,D=2,B=17>            ; 75
   1239 001602' 442007  400340
   1240 001603' 007600  000000             MWORD  <CONT,SOA,OR,D=2,A=16,B=17>        ; 76
   1241 001604' 432167  400340
   1242 001605' 007700  000000             MWORD  <CONT,SOA,AND,D=2,B=0>             ; 77
   1243 001606' 442000  000340
   1244 001607' 010000  000000             MWORD  <CONT,SOA,OR,D=2,A=17,B=0>         ; 100
   1245 001610' 432170  000340
   1246
   1247 001611' 010100  000000             MWORD  <CONT,SOA,AND,D=2,B=1>             ; 101
   1248 001612' 442000  400340
   1249 001613' 010200  000000             MWORD  <CONT,SOA,XOR,D=2,A=0,B=1>         ; 102
   1250 001614' 462000  400340
   1251 001615' 010300  000000             MWORD  <CONT,SOA,AND,D=2,B=2>             ; 103
   1252 001616' 442001  000340
   1253 001617' 010400  000000             MWORD  <CONT,SOA,XOR,D=2,A=1,B=2>         ; 104
   1254 001620' 462011  000340
   1255 001621' 010500  000000             MWORD  <CONT,SOA,AND,D=2,B=3>             ; 105
   1256 001622' 442001  400340
   1257 001623' 010600  000000             MWORD  <CONT,SOA,XOR,D=2,A=2,B=3>         ; 106
   1258 001624' 462021  400340
   1259 001625' 010700  000000             MWORD  <CONT,SOA,AND,D=2,B=4>             ; 107
   1260 001626' 442002  000340
   1261 001627' 011000  000000             MWORD  <CONT,SOA,XOR,D=2,A=3,B=4>         ; 110
   1262 001630' 462032  000340
   1263 001631' 011100  000000             MWORD  <CONT,SOA,AND,D=2,B=5>             ; 111
   1264 001632' 442002  400340
   1265 001633' 011200  000000             MWORD  <CONT,SOA,XOR,D=2,A=4,B=5>         ; 112
   1266 001634' 462042  400340
   1267 001635' 011300  000000             MWORD  <CONT,SOA,AND,D=2,B=6>             ; 113
   1268 001636' 442003  000340
   1269 001637' 011400  000000             MWORD  <CONT,SOA,XOR,D=2,A=5,B=6>         ; 114
   1270 001640' 462053  000340
   1271 001641' 011500  000000             MWORD  <CONT,SOA,AND,D=2,B=7>             ; 115
   1272 001642' 442003  400340
   1273 001643' 011600  000000             MWORD  <CONT,SOA,XOR,D=2,A=6,B=7>         ; 116
   1274 001644' 462063  400340
   1275 001645' 011700  000000             MWORD  <CONT,SOA,AND,D=2,B=10>            ; 117
   1276 001646' 442004  000340
```

```
 1277  001647' 012000  000000              MWORD   <CONT,SOA,XOR,D=2,A=7,B=10>              ; 120
 1278  001650' 462074  000340
 1279  001651' 012100  000000              MWORD   <CONT,SOA,AND,D=2,B=11>                  ; 121
 1280  001652' 442004  400340
 1281  001653' 012200  000000              MWORD   <CONT,SOA,XOR,D=2,A=10,B=11>             ; 122
 1282  001654' 462104  400340
 1283  001655' 012300  000000              MWORD   <CONT,SOA,AND,D=2,B=12>                  ; 123
 1284  001656' 442005  000340
 1285  001657' 012400  000000              MWORD   <CONT,SOA,XOR,D=2,A=11,B=12>             ; 124
 1286  001660' 462115  000340
 1287  001661' 012500  000000              MWORD   <CONT,SOA,AND,D=2,B=13>                  ; 125
 1288  001662' 442005  400340
 1289  001663' 012600  000000              MWORD   <CONT,SOA,XOR,D=2,A=12,B=13>             ; 126
 1290  001664' 462125  400340
 1291  001665' 012700  000000              MWORD   <CONT,SOA,AND,D=2,B=14>                  ; 127
 1292  001666' 442006  000340
 1293  001667' 013000  000000              MWORD   <CONT,SOA,XOR,D=2,A=13,B=14>             ; 130
 1294  001670' 462136  000340
 1295  001671' 013100  000000              MWORD   <CONT,SOA,AND,D=2,B=15>                  ; 131
 1296  001672' 442006  400340
 1297  001673' 013200  000000              MWORD   <CONT,SOA,XOR,D=2,A=14,B=15>             ; 132
 1298  001674' 462146  400340
 1299  001675' 013300  000000              MWORD   <CONT,SOA,AND,D=2,B=16>                  ; 133
 1300  001676' 442007  000340
 1301  001677' 013400  000000              MWORD   <CONT,SOA,XOR,D=2,A=15,B=16>             ; 134
 1302  001700' 462157  000340
 1303  001701' 013500  000000              MWORD   <CONT,SOA,AND,D=2,B=17>                  ; 135
 1304  001702' 442007  400340
 1305  001703' 013600  000000              MWORD   <CONT,SOA,XOR,D=2,A=16,B=17>             ; 136
 1306  001704' 462167  400340
 1307  001705' 013700  000000              MWORD   <CONT,SOA,AND,D=2,B=0>                   ; 137
 1308  001706' 442000  000340
 1309  001707' 014000  000000              MWORD   <CONT,SOA,XOR,D=2,A=17,B=0>              ; 140
 1310  001710' 462170  000340
 1311
 1312  001711' 014100  000000              MWORD   <CONT,SOA,AND,D=2,B=1>                   ; 141
 1313  001712' 442000  400340
 1314  001713' 014200  000000              MWORD   <CONT,SOA,NAND,D=2,A=0,B=1>              ; 142
 1315  001714' 452000  400340
 1316  001715' 014300  000000              MWORD   <CONT,SOA,AND,D=2,B=2>                   ; 143
 1317  001716' 442001  000340
 1318  001717' 014400  000000              MWORD   <CONT,SOA,NAND,D=2,A=1,B=2>              ; 144
 1319  001720' 452011  000340
 1320  001721' 014500  000000              MWORD   <CONT,SOA,AND,D=2,B=3>                   ; 145
 1321  001722' 442001  400340
 1322  001723' 014600  000000              MWORD   <CONT,SOA,NAND,D=2,A=2,B=3>              ; 146
 1323  001724' 452021  400340
 1324  001725' 014700  000000              MWORD   <CONT,SOA,AND,D=2,B=4>                   ; 147
 1325  001726' 442002  000340
 1326  001727' 015000  000000              MWORD   <CONT,SOA,NAND,D=2,A=3,B=4>              ; 150
 1327  001730' 452032  000340
 1328  001731' 015100  000000              MWORD   <CONT,SOA,AND,D=2,B=5>                   ; 151
 1329  001732' 442002  400340
 1330  001733' 015200  000000              MWORD   <CONT,SOA,NAND,D=2,A=4,B=5>              ; 152
 1331  001734' 452042  400340
```

```
  1332  001735' 015300  000000                  MWORD   <CONT,SOA,AND,D=2,B=6>                  ; 153
  1333  001736' 442003  000340
  1334  001737' 015400  000000                  MWORD   <CONT,SOA,NAND,D=2,A=5,B=6>             ; 154
  1335  001740' 452053  000340
  1336  001741' 015500  000000                  MWORD   <CONT,SOA,AND,D=2,B=7>                  ; 155
  1337  001742' 442003  400340
  1338  001743' 015600  000000                  MWORD   <CONT,SOA,NAND,D=2,A=6,B=7>             ; 156
  1339  001744' 452063  400340
  1340  001745' 015700  000000                  MWORD   <CONT,SOA,AND,D=2,B=10>                 ; 157
  1341  001746' 442004  000340
  1342  001747' 016000  000000                  MWORD   <CONT,SOA,NAND,D=2,A=7,B=10>            ; 160
  1343  001750' 452074  000340
  1344  001751' 016100  000000                  MWORD   <CONT,SOA,AND,D=2,B=11>                 ; 161
  1345  001752' 442004  400340
  1346  001753' 016200  000000                  MWORD   <CONT,SOA,NAND,D=2,A=10,B=11>           ; 162
  1347  001754' 452104  400340
  1348  001755' 016300  000000                  MWORD   <CONT,SOA,AND,D=2,B=12>                 ; 163
  1349  001756' 442005  000340
  1350  001757' 016400  000000                  MWORD   <CONT,SOA,NAND,D=2,A=11,B=12>           ; 164
  1351  001760' 452115  000340
  1352  001761' 016500  000000                  MWORD   <CONT,SOA,AND,D=2,B=13>                 ; 165
  1353  001762' 442005  400340
  1354  001763' 016600  000000                  MWORD   <CONT,SOA,NAND,D=2,A=12,B=13>           ; 166
  1355  001764' 452125  400340
  1356  001765' 016700  000000                  MWORD   <CONT,SOA,AND,D=2,B=14>                 ; 167
  1357  001766' 442006  000340
  1358  001767' 017000  000000                  MWORD   <CONT,SOA,NAND,D=2,A=13,B=14>           ; 170
  1359  001770' 452136  000340
  1360  001771' 017100  000000                  MWORD   <CONT,SOA,AND,D=2,B=15>                 ; 171
  1361  001772' 442006  400340
  1362  001773' 017200  000000                  MWORD   <CONT,SOA,NAND,D=2,A=14,B=15>           ; 172
  1363  001774' 452146  400340
  1364  001775' 017300  000000                  MWORD   <CONT,SOA,AND,D=2,B=16>                 ; 173
  1365  001776' 442007  000340
  1366  001777' 017400  000000                  MWORD   <CONT,SOA,NAND,D=2,A=15,B=16>           ; 174
  1367  002000' 452157  000340
  1368  002001' 017500  000000                  MWORD   <CONT,SOA,AND,D=2,B=17>                 ; 175
  1369  002002' 442007  400340
  1370  002003' 017600  000000                  MWORD   <CONT,SOA,NAND,D=2,A=16,B=17>           ; 176
  1371  002004' 452167  400340
  1372  002005' 017700  000000                  MWORD   <CONT,SOA,AND,D=2,B=0>                  ; 177
  1373  002006' 442000  000340
  1374  002007' 020000  000000                  MWORD   <CONT,SOA,NAND,D=2,A=17,B=0>            ; 200
  1375  002010' 452170  000340
  1376
  1377  002011' 020102  022004                  MWORD   <JMAP,J=202,SOA,A=0,OR,OENA,SELE,MGC=4> ; 201
  1378  002012' 430000  005040
  1379  002013' 020202  020000                  MWORD   <JMAP,J=202>                            ; 202
  1380  002014' 000000  000040
  1381  002015' 777777  777777                  -1
```

```

                                      ;#*******************************************************************
                                      ;* TEST 37 - RAM/Q-Register Data Movement Test
                                      ;
                                      ; Description:  Verify that data can be successfully moved to and
                                      ;               from RAM location and Q-Register at full speed.
                                      ;
                                      ; Procedure:    Clear Port
                                      ;               Load microcode
                                      ;
                                      ;               Set RAR to 0
                                      ;               Execute JMAP - load data pattern into Reg 0
                                      ;               Execute a sequence of instructions that moves
                                      ;                 the data from register to register while
                                      ;                 clearing other registers.
                                      ;
                                      ;               When done, read the EBUF and verify that the
                                      ;                 data is correct (516000,,001234).
                                      ;
                                      ; Failure:      ---
                                      ;#*******************************************************************

                                      ; Test data

002016' 254 00 0 00 002025'           TSTA37: JRST    TG37                    ; go start test
002017' 420402  000037                        EBUS!ALU!NDMP!ZALU!37            ; test mask
002020' 002071' 003540'                       T37M,,[ASCIZ ^RAM/Q-Register Data Movement Test^]
002021' 003416' 000000                        [EXP MLAST!E23],,0
002022' 000000  002244'                       TSTA40                          ; failure test table
002023' 000000  002467'                       TSTA41                          ;   ...
002024' 777777  777777                        -1

                                      ; Start test

002025' 201 00 0 00 000000'           TG37:   MOVEI   Z7                      ; get address of module start
002026' 260 17 0 00 001344*                   GO      TRACE                   ; handle trace output
002027' 201 01 0 00 002071'                   MOVEI   1,T37M                  ; set up microcode address
002030' 260 17 0 00 001346*                   GO      TLOAD                   ; load/verify it
002031' 263 17 0 00 000000                    RTN                             ; failed - exit test

                                      ; Initialization

002032' 400 15 0 00 000000            TL37:   SETZ    ERFLG,                  ; clear error flag
002033' 260 17 0 00 001351*                   GO      IPACLR                  ; clear port
```

```


                                       ; 1st segment of test (Segment A) - Start up port and wait till it stops

002034' 201 01 0 00 000012           TA37:    MOVEI   1,^D10                     ; maximum delay before timeout (msec)
002035' 201 02 0 00 000000                    MOVEI   2,0                        ; error address
002036' 201 03 0 00 000032                    MOVEI   3,32                       ; correct address
002037' 400 04 0 00 000000                    SETZ    4,                         ; starting address
002040' 260 17 0 00 001356*                   GO      AEXEC                      ; execute ALU type test
002041' 255 00 0 00 000000                    JFCL                               ; ignore error
002042' 400 15 0 00 000000                    SETZ    ERFLG,                     ; clear error flag
002043' 332 00 0 00 001361*                   SKIPE   ALFLS                      ; started up ok?
002044' 474 15 0 00 000000                    SETO    ERFLG,                     ; no - set error flag
002045' 200 00 0 00 001402*                   MOVE    ALEBF                      ; get EBUF data
002046' 404 00 0 00 003514'                   AND     [777400,,1777]             ; mask out indeterminate bits
002047' 312 00 0 00 003515'                   CAME    [516000,,1234]             ; correct?
002050' 474 15 0 00 000000                    SETO    ERFLG,                     ; no - set error flag

                                       ; Handle error printouts and scope looping

002051' 027 00 0 00 002055'                   SCOPER  MA37                       ; print error message
002052' 254 00 0 00 002032'                   JRST    TL37                       ; loop on error
002053' 254 00 0 00 002054'                   JRST    TX37                       ; altmode exit

                                       ; End of test

002054' 263 17 0 00 000000           TX37:    RTN                                ; return

                                       ; Error messages

002055' 140000  003547'              MA37:    MSG!TXNOT![ASCIZ ^2901 RAM/Q-Register data got destroyed somehow (result in EBUF)^]
002056' 270000  002057'                       LAST!CALL!TXALL!MA37PN

002057' 200 01 0 00 001375*          MA37PN:  MOVE    1,SCOSW                    ; get switches
002060' 603 01 0 00 000200                    TLNE    1,TXTINH                   ; full printout?
                                              TMSGC   <EBUF (C):  000000 000001
002061' 037 00 0 00 003450'                (A):  >
002062' 607 01 0 00 000200                    TLNN    1,TXTINH                   ; full printout?
                                              TMSGC   <EBUF (Correct):  000000 000001
002063' 037 00 0 00 003460'                 (Actual):  >
002064' 200 00 0 00 002045*                   MOVE    ALEBF
002065' 037 13 0 00 000000                    PNTHW
002066' 607 01 0 00 000200                    TLNN    1,TXTINH                   ; full printout?
002067' 037 00 0 00 003531'                   TMSGC   <(Middle 16 bits indeterminate)>
002070' 263 17 0 00 000000                    RTN
```

```

                                ; Microcode:

002071' 000000  011234          T37M:   MWORD   <ADDR=0,JMAP,J=1,SD0,OR,D=2,SKCN,MGC=1234> ; 0
002072' 732000  240040
002073' 000100  100000                  MWORD   <LDCT,J=10,D=1>                         ; 1
002074' 001000  000300
002075' 000200  000000                  MWORD   <CONT,SOA,AND,D=4,B=1>                  ; 2
002076' 444000  400340
002077' 000300  000000                  MWORD   <CONT,SOA,PLUS,D=0,A=0>                 ; 3
002100' 400000  000340
002101' 000400  000000                  MWORD   <CONT,SOQ,PLUS,D=2,B=1>                 ; 4
002102' 202000  400340
002103' 000500  000000                  MWORD   <CONT,SOA,AND,D=4,B=2>                  ; 5
002104' 444001  000340
002105' 000600  000000                  MWORD   <CONT,SOA,OR,D=0,A=1>                   ; 6
002106' 430010  000340
002107' 000700  000000                  MWORD   <CONT,SOQ,OR,D=2,B=2>                   ; 7
002110' 232001  000340
002111' 001000  000000                  MWORD   <CONT,SOA,AND,D=4,B=3>                  ; 10
002112' 444001  400340
002113' 001100  000000                  MWORD   <CONT,SOA,XOR,D=0,A=2>                  ; 11
002114' 460020  000340
002115' 001200  000000                  MWORD   <CONT,SOQ,XOR,D=2,B=3>                  ; 12
002116' 262001  400340
002117' 001300  000000                  MWORD   <CONT,SOA,AND,D=4,B=4>                  ; 13
002120' 444002  000340
002121' 001400  000000                  MWORD   <CONT,SOA,NAND,D=0,A=3>                 ; 14
002122' 450030  000340
002123' 001500  000000                  MWORD   <CONT,SOQ,NAND,D=2,B=4>                 ; 15
002124' 252002  000340
002125' 001600  000000                  MWORD   <CONT,SOA,AND,D=4,B=5>                  ; 16
002126' 444002  400340
002127' 001700  000000                  MWORD   <CONT,SOA,PLUS,D=0,A=4>                 ; 17
002130' 400040  000340
002131' 002000  000000                  MWORD   <CONT,SOQ,PLUS,D=2,B=5>                 ; 20
002132' 202002  400340
002133' 002100  000000                  MWORD   <CONT,SOA,AND,D=4,B=6>                  ; 21
002134' 444003  000340
002135' 002200  000000                  MWORD   <CONT,SOA,OR,D=0,A=5>                   ; 22
002136' 430050  000340
002137' 002300  000000                  MWORD   <CONT,SOQ,OR,D=2,B=6>                   ; 23
002140' 232003  000340
002141' 002400  000000                  MWORD   <CONT,SOA,AND,D=4,B=7>                  ; 24
002142' 444003  400340
002143' 002500  000000                  MWORD   <CONT,SOA,XOR,D=0,A=6>                  ; 25
002144' 460060  000340
002145' 002600  000000                  MWORD   <CONT,SOQ,XOR,D=2,B=7>                  ; 26
002146' 262003  400340
002147' 002700  000000                  MWORD   <CONT,SOA,AND,D=4,B=10>                 ; 27
002150' 444004  000340
002151' 003000  000000                  MWORD   <CONT,SOA,NAND,D=0,A=7>                 ; 30
002152' 450070  000340
002153' 003100  000000                  MWORD   <CONT,SOQ,NAND,D=2,B=10>                ; 31
002154' 252004  000340
```

```
1525  002155' 003200  000000            MWORD   <CONT,SOA,AND,D=4,B=11>                 ; 32
1526  002156' 444004  400340
1527  002157' 003300  000000            MWORD   <CONT,SOA,PLUS,D=0,A=10>                ; 33
1528  002160' 400100  000340
1529  002161' 003400  000000            MWORD   <CONT,SOQ,PLUS,D=2,B=11>                ; 34
1530  002162' 202004  400340
1531  002163' 003500  000000            MWORD   <CONT,SOA,AND,D=4,B=12>                 ; 35
1532  002164' 444005  000340
1533  002165' 003600  000000            MWORD   <CONT,SOA,OR,D=0,A=11>                  ; 36
1534  002166' 430110  000340
1535  002167' 003700  000000            MWORD   <CONT,SOQ,OR,D=2,B=12>                  ; 37
1536  002170' 232005  000340
1537  002171' 004000  000000            MWORD   <CONT,SOA,AND,D=4,B=13>                 ; 40
1538  002172' 444005  400340
1539  002173' 004100  000000            MWORD   <CONT,SOA,XOR,D=0,A=12>                 ; 41
1540  002174' 460120  000340
1541  002175' 004200  000000            MWORD   <CONT,SOQ,XOR,D=2,B=13>                 ; 42
1542  002176' 262005  400340
1543  002177' 004300  000000            MWORD   <CONT,SOA,AND,D=4,B=                    ; 43
1544  002200' 444006  000340
1545  002201' 004400  000000            MWORD   <CONT,SOA,NAND,D=0,A=13>                ; 44
1546  002202' 450130  000340
1547  002203' 004500  000000            MWORD   <CONT,SOQ,NAND,D=2,B=14>                ; 45
1548  002204' 252006  000340
1549  002205' 004600  000000            MWORD   <CONT,SOA,AND,D=4,B=15>                 ; 46
1550  002206' 444006  400340
1551  002207' 004700  000000            MWORD   <CONT,SOA,PLUS,D=0,A=14>                ; 47
1552  002210' 400140  000340
1553  002211' 005000  000000            MWORD   <CONT,SOQ,PLUS,D=2,B=15>                ; 50
1554  002212' 202006  400340
1555  002213' 005100  000000            MWORD   <CONT,SOA,AND,D=4,B=16>                 ; 51
1556  002214' 444007  000340
1557  002215' 005200  000000            MWORD   <CONT,SOA,OR,D=0,A=15>                  ; 52
1558  002216' 430150  000340
1559  002217' 005300  000000            MWORD   <CONT,SOQ,OR,D=2,B=16>                  ; 53
1560  002220' 232007  000340
1561  002221' 005400  000000            MWORD   <CONT,SOA,AND,D=4,B=17>                 ; 54
1562  002222' 444007  400340
1563  002223' 005500  000000            MWORD   <CONT,SOA,XOR,D=0,A=16>                 ; 55
1564  002224' 460160  000340
1565  002225' 005600  000000            MWORD   <CONT,SOQ,XOR,D=2,B=17>                 ; 56
1566  002226' 262007  400340
1567  002227' 005700  000000            MWORD   <CONT,SOA,AND,D=4,B=0>                  ; 57
1568  002230' 444000  000340
1569  002231' 006000  000000            MWORD   <CONT,SOA,NAND,D=0,A=17>                ; 60
1570  002232' 450170  000340
1571  002233' 006100  000000            MWORD   <CONT,SOQ,NAND,D=2,A=17,B=0>            ; 61
1572  002234' 252170  000340
1573  002235' 006200  020000            MWORD   <RPCT,J=2,D=1>                          ; 62
1574  002236' 001000  000220
1575
1576  002237' 006300  642004            MWORD   <JMAP,J=64,SOA,A=0,OR,OENA,SELE,MGC=4>  ; 63
1577  002240' 430000  005040
1578  002241' 006400  640000            MWORD   <JMAP,J=64>                             ; 64
1579  002242' 000000  000040
```

```
  1580  002243' 777777  777777                    -1
```

```
                                                  ;#*********************************************************************
                                                  ;* TEST 40 - RAM Data Movement/Shifting Test
                                                  ;
                                                  ; Description:  Verify that data can be successfully moved to and
                                                  ;               from various RAM locations  at  full  speed while
                                                  ;               shifting left and right.
                                                  ;
                                                  ; Procedure:    Clear Port
                                                  ;               Load microcode
                                                  ;
                                                  ;               Set RAR to 0
                                                  ;               Execute JMAP - load data pattern into Reg 0
                                                  ;               Execute a sequence of instructions that moves
                                                  ;                 the data from register to register while
                                                  ;                 clearing other registers and while shifting
                                                  ;                 left and right.
                                                  ;
                                                  ;               When done, read the EBUF and verify that the
                                                  ;                 data is correct (516000,,000000), which is
                                                  ;                 516000,,001234 left shifted 26 bits.
                                                  ;
                                                  ; Failure:      ---
                                                  ;#*********************************************************************

                                                  ; Test data

        002244' 254 00 0 00 002252'               TSTA40: JRST    TG40                        ; go start test
        002245' 420402  000040                            EBUS!ALU!NDMP!ZALU!40               ; test mask
        002246' 002314' 003564'                           T40M,,[ASCIZ ^RAM Data Movement/Shifting Test^]
        002247' 003416' 000000                            [EXP MLAST!E23],,0
        002250' 000000  002467'                           TSTA41                              ; failure test table
        002251' 777777  777777                            -1                                  ;   ...

                                                  ; Start test

        002252' 201 00 0 00 000000'               TG40:   MOVEI   Z7                          ; get address of module start
        002253' 260 17 0 00 002026*                       GO      TRACE                       ; handle trace output
        002254' 201 01 0 00 002314'                       MOVEI   1,T40M                      ; set up microcode address
        002255' 260 17 0 00 002030*                       GO      TLOAD                       ; load/verify it
        002256' 263 17 0 00 000000                        RTN                                 ; failed - exit test

                                                  ; Initialization

        002257' 400 15 0 00 000000                TL40:   SETZ    ERFLG,                      ; clear error flag
        002260' 260 17 0 00 002033*                       GO      IPACLR                      ; clear port
```

```

                                  ; 1st segment of test (Segment A) - Start up port and wait till it stops

002261' 201 01 0 00 000012        TA40:   MOVEI   1,^D10                  ; maximum delay before timeout (msec)
002262' 201 02 0 00 000000                MOVEI   2,0                     ; error address
002263' 201 03 0 00 000032                MOVEI   3,32                    ; correct address
002264' 400 04 0 00 000000                SETZ    4,                      ; starting address
002265' 260 17 0 00 002040*               GO      AEXEC                   ; execute ALU type test
002266' 255 00 0 00 000000                JFCL                            ; ignore error
002267' 400 15 0 00 000000                SETZ    ERFLG,                  ; clear error flag
002270' 332 00 0 00 002043*               SKIPE   ALFLS                   ; started up ok?
002271' 474 15 0 00 000000                SETO    ERFLG,                  ; no - set error flag
002272' 200 00 0 00 002064*               MOVE    ALEBF                   ; get EBUF data
002273' 404 00 0 00 003514'               AND     [777400,,1777]          ; mask out indeterminate bits
002274' 312 00 0 00 003573'               CAME    [516000,,0]             ; correct?
002275' 474 15 0 00 000000                SETO    ERFLG,                  ; no - set error flag

                                  ; Handle error printouts and scope looping

002276' 027 00 0 00 002302'               SCOPER  MA40                    ; print error message
002277' 254 00 0 00 002257'               JRST    TL40                    ; loop on error
002300' 254 00 0 00 002301'               JRST    TX40                    ; altmode exit

                                  ; End of test

002301' 263 17 0 00 000000        TX40:   RTN                             ; return

                                  ; Error messages

002302' 140000  003574'           MA40:   MSG!TXNOT![ASCIZ ^2901 RAM Register data got destroyed somehow^]
002303' 270000  002304'                   LAST!CALL!TXALL!MA40PN

002304' 200 01 0 00 002057*       MA40PN: MOVE    1,SCOSW                 ; get switches
002305' 603 01 0 00 000200                TLNE    1,TXTINH                ; full printout?
                                          TMSGC   <EBUF (C):  516000 000000
002306' 037 00 0 00 003605'           (A):  >
002307' 607 01 0 00 000200                TLNN    1,TXTINH                ; full printout?
                                          TMSGC   <EBUF (Correct):  516000 000000
002310' 037 00 0 00 003615'           (Actual):  >
002311' 200 00 0 00 002272*               MOVE    ALEBF
002312' 037 13 0 00 000000                PNTHW
002313' 263 17 0 00 000000                RTN
```

```


                                   ; Microcode:

002314' 000000  011234             T40M:    MWORD   <ADDR=0,JMAP,J=1,SD0,OR,D=2,SKCN,MGC=1234> ; 0
002315' 732000  240040
002316' 000100  310000                      MWORD   <LDCT,J=31,D=1>                   ; 1
002317' 001000  000300
002320' 000200  000000                      MWORD   <CONT,SOA,AND,D=4,B=1>            ; 2
002321' 444000  400340
002322' 000300  000000                      MWORD   <CONT,SOB,PLUS,D=7,B=0>           ; 3
002323' 307000  000340
002324' 000400  000000                      MWORD   <CONT,SOA,PLUS,D=5,A=0,B=1>       ; 4
002325' 405000  400340
002326' 000500  000000                      MWORD   <CONT,SOA,AND,D=4,B=2>            ; 5
002327' 444001  000340
002330' 000600  000000                      MWORD   <CONT,SOB,OR,D=7,B=1>             ; 6
002331' 337000  400340
002332' 000700  000000                      MWORD   <CONT,SOA,OR,D=5,B=2>             ; 7
002333' 435001  000340
002334' 001000  000000                      MWORD   <CONT,SOA,AND,D=4,B=3>            ; 10
002335' 444001  400340
002336' 001100  000000                      MWORD   <CONT,SOB,XOR,D=7,B=2>            ; 11
002337' 367001  000340
002340' 001200  000000                      MWORD   <CONT,SOA,XOR,D=5,B=3>            ; 12
002341' 465001  400340
002342' 001300  000000                      MWORD   <CONT,SOA,AND,D=4,B=4>            ; 13
002343' 444002  000340
002344' 001400  000000                      MWORD   <CONT,SOB,NAND,D=7,B=3>           ; 14
002345' 357001  400340
002346' 001500  000000                      MWORD   <CONT,SOA,NAND,D=5,B=4>           ; 15
002347' 455002  000340
002350' 001600  000000                      MWORD   <CONT,SOA,AND,D=4,B=5>            ; 16
002351' 444002  400340
002352' 001700  000000                      MWORD   <CONT,SOB,PLUS,D=7,B=4>           ; 17
002353' 307002  000340
002354' 002000  000000                      MWORD   <CONT,SOA,PLUS,D=5,B=5>           ; 20
002355' 405002  400340
002356' 002100  000000                      MWORD   <CONT,SOA,AND,D=4,B=6>            ; 21
002357' 444003  000340
002360' 002200  000000                      MWORD   <CONT,SOB,OR,D=7,B=5>             ; 22
002361' 337002  400340
002362' 002300  000000                      MWORD   <CONT,SOA,OR,D=5,B=6>             ; 23
002363' 435003  000340
002364' 002400  000000                      MWORD   <CONT,SOA,AND,D=4,B=7>            ; 24
002365' 444003  400340
002366' 002500  000000                      MWORD   <CONT,SOB,XOR,D=7,B=6>            ; 25
002367' 367003  000340
002370' 002600  000000                      MWORD   <CONT,SOA,XOR,D=5,B=7>            ; 26
002371' 465003  400340
002372' 002700  000000                      MWORD   <CONT,SOA,AND,D=4,B=10>           ; 27
002373' 444004  000340
002374' 003000  000000                      MWORD   <CONT,SOB,NAND,D=7,B=7>           ; 30
002375' 357003  400340
002376' 003100  000000                      MWORD   <CONT,SOA,NAND,D=5,B=10>          ; 31
002377' 455004  000340
```

```
002400' 003200  000000          MWORD   <CONT,SOA,AND,D=4,B=11>                 ; 32
002401' 444004  400340
002402' 003300  000000          MWORD   <CONT,SOB,PLUS,D=7,B=10>                ; 33
002403' 307004  000340
002404' 003400  000000          MWORD   <CONT,SOA,PLUS,D=5,B=11>                ; 34
002405' 405004  400340
002406' 003500  000000          MWORD   <CONT,SOA,AND,D=4,B=12>                 ; 35
002407' 444005  000340
002410' 003600  000000          MWORD   <CONT,SOB,OR,D=7,B=11>                  ; 36
002411' 337004  400340
002412' 003700  000000          MWORD   <CONT,SOA,OR,D=5,B=12>                  ; 37
002413' 435005  000340
002414' 004000  000000          MWORD   <CONT,SOA,AND,D=4,B=13>                 ; 40
002415' 444005  400340
002416' 004100  000000          MWORD   <CONT,SOB,XOR,D=7,B=12>                 ; 41
002417' 367005  000340
002420' 004200  000000          MWORD   <CONT,SOA,XOR,D=5,B=13>                 ; 42
002421' 465005  400340
002422' 004300  000000          MWORD   <CONT,SOA,AND,D=4,B=14>                 ; 43
002423' 444006  000340
002424' 004400  000000          MWORD   <CONT,SOB,NAND,D=7,B=13>                ; 44
002425' 357005  400340
002426' 004500  000000          MWORD   <CONT,SOA,NAND,D=5,B=14>                ; 45
002427' 455006  000340
002430' 004600  000000          MWORD   <CONT,SOA,AND,D=4,B=15>                 ; 46
002431' 444006  400340
002432' 004700  000000          MWORD   <CONT,SOB,PLUS,D=7,B=14>                ; 47
002433' 307006  000340
002434' 005000  000000          MWORD   <CONT,SOA,PLUS,D=5,B=15>                ; 50
002435' 405006  400340
002436' 005100  000000          MWORD   <CONT,SOA,AND,D=4,B=16>                 ; 51
002437' 444007  000340
002440' 005200  000000          MWORD   <CONT,SOB,OR,D=7,B=15>                  ; 52
002441' 337006  400340
002442' 005300  000000          MWORD   <CONT,SOA,OR,D=5,B=16>                  ; 53
002443' 435007  000340
002444' 005400  000000          MWORD   <CONT,SOA,AND,D=4,B=17>                 ; 54
002445' 444007  400340
002446' 005500  000000          MWORD   <CONT,SOB,XOR,D=7,B=16>                 ; 55
002447' 367007  000340
002450' 005600  000000          MWORD   <CONT,SOA,XOR,D=5,B=17>                 ; 56
002451' 465007  400340
002452' 005700  000000          MWORD   <CONT,SOA,AND,D=4,B=0>                  ; 57
002453' 444000  000340
002454' 006000  000000          MWORD   <CONT,SOB,NAND,D=7,B=17>                ; 60
002455' 357007  400340
002456' 006100  000000          MWORD   <CONT,SOA,NAND,D=5,A=17,B=0>            ; 61
002457' 455170  000340
002460' 006200  020000          MWORD   <RPCT,J=2,SOB,OR,B=0,D=7>               ; 62
002461' 337000  000220

002462' 006300  642004          MWORD   <JMAP,J=64,SOA,A=0,OR,OENA,SELE,MGC=4>  ; 63
002463' 430000  005040
002464' 006400  640000          MWORD   <JMAP,J=64>                             ; 64
002465' 000000  000040
```

```
  1779  002466' 777777  777777                -1
```

```
                                        ;***********************************************************************
                                        ;* TEST 41 - RAM/Q-Reg Data Movement/Shifting Test
                                        ;
                                        ; Description:  Verify that data can be successfully moved to and
                                        ;               from RAM location and Q-Register  at  full  speed
                                        ;               while shifting left and right.
                                        ;
                                        ; Procedure:    Clear Port
                                        ;               Load microcode
                                        ;
                                        ;               Set RAR to 0
                                        ;               Execute JMAP - load data pattern into Reg 0
                                        ;               Execute a sequence of instructions that moves
                                        ;                 the data from register to register while
                                        ;                 clearing other registers and while shifting
                                        ;                 left and right.
                                        ;
                                        ;               When done, read the EBUF and verify that the
                                        ;                 data is correct (516000,,000000), which is
                                        ;                 516000,,001234 left shifted 26 bits.
                                        ;
                                        ; Failure:      ---
                                        ;***********************************************************************

                                        ; Test data

002467' 254 00 0 00 002474'             TSTA41: JRST    TG41                    ; go start test
002470' 420402  000041                          EBUS!ALU!NDMP!ZALU!41           ; test mask
002471' 002536' 003630'                         T41M,,[ASCIZ ^RAM/Q-Reg Data Movement/Shifting Test^]
002472' 003416' 000000                          [EXP MLAST!E23],,0
002473' 777777  777777                          -1                              ; failure test table

                                        ; Start test

002474' 201 00 0 00 000000'             TG41:   MOVEI   Z7                      ; get address of module start
002475' 260 17 0 00 002253*                     GO      TRACE                   ; handle trace output
002476' 201 01 0 00 002536'                     MOVEI   1,T41M                  ; set up microcode address
002477' 260 17 0 00 002255*                     GO      TLOAD                   ; load/verify it
002500' 263 17 0 00 000000                      RTN                             ; failed - exit test

                                        ; Initialization

002501' 400 15 0 00 000000              TL41:   SETZ    ERFLG,                  ; clear error flag
002502' 260 17 0 00 002260*                     GO      IPACLR                  ; clear port
```

```

                                        ; 1st segment of test (Segment A) - Start up port and wait till it stops

002503' 201 01 0 00 000012      TA41:   MOVEI   1,^D10                  ; maximum delay before timeout (msec)
002504' 201 02 0 00 000000              MOVEI   2,0                     ; error address
002505' 201 03 0 00 000032              MOVEI   3,32                    ; correct address
002506' 400 04 0 00 000000              SETZ    4,                      ; starting address
002507' 260 17 0 00 002265*             GO      AEXEC                   ; execute ALU type test
002510' 255 00 0 00 000000              JFCL                            ; ignore error
002511' 400 15 0 00 000000              SETZ    ERFLG,                  ; clear error flag
002512' 332 00 0 00 002270*             SKIPE   ALFLS                   ; started up ok?
002513' 474 15 0 00 000000              SETO    ERFLG,                  ; no - set error flag
002514' 200 00 0 00 002311*             MOVE    ALEBF                   ; get EBUF data
002515' 404 00 0 00 003514'             AND     [777400,,1777]          ; mask out indeterminate bits
002516' 312 00 0 00 003573'             CAME    [516000,,0]             ; correct?
002517' 474 15 0 00 000000              SETO    ERFLG,                  ; no - set error flag

                                ; Handle error printouts and scope looping

002520' 027 00 0 00 002524'             SCOPER  MA41                    ; print error message
002521' 254 00 0 00 002501'             JRST    TL41                    ; loop on error
002522' 254 00 0 00 002523'             JRST    TX41                    ; altmode exit

                                ; End of test

002523' 263 17 0 00 000000      TX41:   RTN                             ; return

                                ; Error messages

002524' 160000  003640'         MA41:   MSG!TXALL![ASCIZ ^2901 RAM/Q-Register data got destroyed somehow^]
002525' 270000  002526'                 LAST!CALL!TXALL!MA41PN

002526' 200 01 0 00 002304*     MA41PN: MOVE    1,SCOSW                 ; get switches
002527' 603 01 0 00 000200              TLNE    1,TXTINH                ; full printout?
                                        TMSGC   <EBUF (C):  516000 000000
002530' 037 00 0 00 003605'           (A):  >
002531' 607 01 0 00 000200              TLNN    1,TXTINH                ; full printout?
                                        TMSGC   <EBUF (Correct):  516000 000000
002532' 037 00 0 00 003615'            (Actual):  >
002533' 200 00 0 00 002514*             MOVE    ALEBF
002534' 037 13 0 00 000000              PNTHW
002535' 263 17 0 00 000000              RTN
```

```
                                        ; Microcode:

002536' 000000  011234          T41M:   MWORD   <ADDR=0,JMAP,J=1,SD0,OR,D=2,SKCN,MGC=1234> ; 0
002537' 732000  240040
002540' 000100  310000                  MWORD   <LDCT,J=31,D=1>                          ; 1
002541' 001000  000300
002542' 000200  000000                  MWORD   <CONT,SOA,AND,D=4,B=1>                   ; 2
002543' 444000  400340
002544' 000300  000000                  MWORD   <CONT,SOA,PLUS,D=6,A=0,B=1>              ; 3
002545' 406000  400340
002546' 000400  000000                  MWORD   <CONT,SOQ,PLUS,D=5,B=1>                  ; 4
002547' 205000  400340
002550' 000500  000000                  MWORD   <CONT,SOA,AND,D=4,B=2>                   ; 5
002551' 444001  000340
002552' 000600  000000                  MWORD   <CONT,SOA,OR,D=6,A=1,B=1>                ; 6
002553' 436010  400340
002554' 000700  000000                  MWORD   <CONT,SOQ,OR,D=5,B=2>                    ; 7
002555' 235001  000340
002556' 001000  000000                  MWORD   <CONT,SOA,AND,D=4,B=3>                   ; 10
002557' 444001  400340
002560' 001100  000000                  MWORD   <CONT,SOA,XOR,D=6,A=2,B=1>               ; 11
002561' 466020  400340
002562' 001200  000000                  MWORD   <CONT,SOQ,XOR,D=5,B=3>                   ; 12
002563' 265001  400340
002564' 001300  000000                  MWORD   <CONT,SOA,AND,D=4,B=4>                   ; 13
002565' 444002  000340
002566' 001400  000000                  MWORD   <CONT,SOA,NAND,D=6,A=3,B=1>              ; 14
002567' 456030  400340
002570' 001500  000000                  MWORD   <CONT,SOQ,NAND,D=5,B=4>                  ; 15
002571' 255002  000340
002572' 001600  000000                  MWORD   <CONT,SOA,AND,D=4,B=5>                   ; 16
002573' 444002  400340
002574' 001700  000000                  MWORD   <CONT,SOA,PLUS,D=6,A=4,B=1>              ; 17
002575' 406040  400340
002576' 002000  000000                  MWORD   <CONT,SOQ,PLUS,D=5,B=5>                  ; 20
002577' 205002  400340
002600' 002100  000000                  MWORD   <CONT,SOA,AND,D=4,B=6>                   ; 21
002601' 444003  000340
002602' 002200  000000                  MWORD   <CONT,SOA,OR,D=6,A=5,B=1>                ; 22
002603' 436050  400340
002604' 002300  000000                  MWORD   <CONT,SOQ,OR,D=5,B=6>                    ; 23
002605' 235003  000340
002606' 002400  000000                  MWORD   <CONT,SOA,AND,D=4,B=7>                   ; 24
002607' 444003  400340
002610' 002500  000000                  MWORD   <CONT,SOA,XOR,D=6,A=6,B=1>               ; 25
002611' 466060  400340
002612' 002600  000000                  MWORD   <CONT,SOQ,XOR,D=5,B=7>                   ; 26
002613' 265003  400340
002614' 002700  000000                  MWORD   <CONT,SOA,AND,D=4,B=10>                  ; 27
002615' 444004  000340
002616' 003000  000000                  MWORD   <CONT,SOA,NAND,D=6,A=7,B=1>              ; 30
002617' 456070  400340
002620' 003100  000000                  MWORD   <CONT,SOQ,NAND,D=5,B=10>                 ; 31
002621' 255004  000340
```

```
 1922  002622' 003200  000000          MWORD   <CONT,SOA,AND,D=4,B=11>                 ; 32
 1923  002623' 444004  400340
 1924  002624' 003300  000000          MWORD   <CONT,SOA,PLUS,D=6,A=10,B=1>            ; 33
 1925  002625' 406100  400340
 1926  002626' 003400  000000          MWORD   <CONT,SOQ,PLUS,D=5,B=11>                ; 34
 1927  002627' 205004  400340
 1928  002630' 003500  000000          MWORD   <CONT,SOA,AND,D=4,B=12>                 ; 35
 1929  002631' 444005  000340
 1930  002632' 003600  000000          MWORD   <CONT,SOA,OR,D=6,A=11,B=1>              ; 36
 1931  002633' 436110  400340
 1932  002634' 003700  000000          MWORD   <CONT,SOQ,OR,D=5,B=12>                  ; 37
 1933  002635' 235005  000340
 1934  002636' 004000  000000          MWORD   <CONT,SOA,AND,D=4,B=13>                 ; 40
 1935  002637' 444005  400340
 1936  002640' 004100  000000          MWORD   <CONT,SOA,XOR,D=6,A=12,B=1>             ; 41
 1937  002641' 466120  400340
 1938  002642' 004200  000000          MWORD   <CONT,SOQ,XOR,D=5,B=13>                 ; 42
 1939  002643' 265005  400340
 1940  002644' 004300  000000          MWORD   <CONT,SOA,AND,D=4,B=14>                 ; 43
 1941  002645' 444006  000340
 1942  002646' 004400  000000          MWORD   <CONT,SOA,NAND,D=6,A=13,B=1>            ; 44
 1943  002647' 456130  400340
 1944  002650' 004500  000000          MWORD   <CONT,SOQ,NAND,D=5,B=14>                ; 45
 1945  002651' 255006  000340
 1946  002652' 004600  000000          MWORD   <CONT,SOA,AND,D=4,B=15>                 ; 46
 1947  002653' 444006  400340
 1948  002654' 004700  000000          MWORD   <CONT,SOA,PLUS,D=6,A=14,B=1>            ; 47
 1949  002655' 406140  400340
 1950  002656' 005000  000000          MWORD   <CONT,SOQ,PLUS,D=5,B=15>                ; 50
 1951  002657' 205006  400340
 1952  002660' 005100  000000          MWORD   <CONT,SOA,AND,D=4,B=16>                 ; 51
 1953  002661' 444007  000340
 1954  002662' 005200  000000          MWORD   <CONT,SOA,OR,D=6,A=15,B=1>              ; 52
 1955  002663' 436150  400340
 1956  002664' 005300  000000          MWORD   <CONT,SOQ,OR,D=5,B=16>                  ; 53
 1957  002665' 235007  000340
 1958  002666' 005400  000000          MWORD   <CONT,SOA,AND,D=4,B=17>                 ; 54
 1959  002667' 444007  400340
 1960  002670' 005500  000000          MWORD   <CONT,SOA,XOR,D=6,A=16,B=1>             ; 55
 1961  002671' 466160  400340
 1962  002672' 005600  000000          MWORD   <CONT,SOQ,XOR,D=5,B=17>                 ; 56
 1963  002673' 265007  400340
 1964  002674' 005700  000000          MWORD   <CONT,SOA,AND,D=4,B=1>                  ; 57
 1965  002675' 444000  400340
 1966  002676' 006000  000000          MWORD   <CONT,SOA,NAND,D=6,A=17,B=1>            ; 60
 1967  002677' 456170  400340
 1968  002700' 006100  000000          MWORD   <CONT,SOQ,NAND,D=5,A=17,B=1>            ; 61
 1969  002701' 255170  400340
 1970  002702' 006200  020000          MWORD   <RPCT,J=2,SOB,OR,B=0,D=7>               ; 62
 1971  002703' 337000  000220
 1972
 1973  002704' 006300  632004          MWORD   <JMAP,J=63,SOA,A=0,OR,D=1,OENA,SELE,MGC=4> ; 63
 1974  002705' 431000  005040
 1975  002706' 777777  777777          -1
```

$B

```

                                        ;#*********************************************************************
                                        ;* TEST 42 - 2901 Register / Constant Mux Test
                                        ;
                                        ; Description:  Verify the AND function  of  the  2901 registers,
                                        ;               anding  them  with data on the direct inputs (the
                                        ;               Constant MUX).
                                        ;
                                        ; Procedure:    Clear Port
                                        ;               Load microcode
                                        ;
                                        ;               Write 0's to register x
                                        ;               Write 1's to register y
                                        ;               And a constant of 1's with register y and place in
                                        ;                 register x.
                                        ;               Verify that register x now contains all zero's.
                                        ;
                                        ;               Repeat with various register combinations.
                                        ;
                                        ; Failure:      ---
                                        ;#*********************************************************************

                                        ; Test data

002707' 254 00 0 00 002714'             TSTA42: JRST    TG42                    ; go start test
002710' 420402  000042                          EBUS!ALU!NDMP!ZALU!42           ; test mask
002711' 002737' 003652'                         T42M,,[ASCIZ ^2901 Register / Constant Mux Test^]
002712' 003416' 000000                          [EXP MLAST!E23],,0
002713' 777777  777777                          -1                              ; failure test table

                                        ; Start test

002714' 201 00 0 00 000000'             TG42:   MOVEI   Z7                      ; get address of module start
002715' 260 17 0 00 002475*                     GO      TRACE                   ; handle trace output
002716' 201 01 0 00 002737'                     MOVEI   1,T42M                  ; set up microcode address
002717' 260 17 0 00 002477*                     GO      TLOAD                   ; load/verify it
002720' 263 17 0 00 000000                      RTN                             ; failed - exit test

                                        ; Initialization

002721' 400 15 0 00 000000              TL42:   SETZ    ERFLG,                  ; clear error flag
002722' 260 17 0 00 002502*                     GO      IPACLR                  ; clear port
```

```

                                        ; 1st segment of test (Segment A) - Start up port and wait till it stops

002723' 201 01 0 00 000012              TA42:   MOVEI   1,^D10                  ; maximum delay before timeout (msec)
002724' 201 02 0 00 005000                      MOVEI   2,5000                  ; error address
002725' 201 03 0 00 004000                      MOVEI   3,4000                  ; correct address
002726' 201 04 0 00 000001                      MOVEI   4,1                     ; starting address
002727' 260 17 0 00 002507*                     GO      AEXEC                   ; execute ALU type test
002730' 474 15 0 00 000000                      SETO    ERFLG,                  ; error occurred

                                        ; Handle error printouts and scope looping

002731' 027 00 0 00 002735'                     SCOPER  MA42                    ; print error message
002732' 254 00 0 00 002721'                     JRST    TL42                    ; loop on error
002733' 254 00 0 00 002734'                     JRST    TX42                    ; altmode exit

                                        ; End of test

002734' 263 17 0 00 000000              TX42:   RTN                             ; return

                                        ; Error message - Microcode did not stop in the right place

002735' 160000  003661'                 MA42:   MSG!TXALL![ASCIZ ^2901 function using Constant MUX failed^]
002736' 000000000000#                           LAST!CALL!TXALL!AAPNT

                                        ; Microcode:

002737' 000001  000000                  T42M:   MWORD   <ADDR=0,JMAP,J=100>                             ; 0
002740' 000000  000040
002741' 000100  000000                          MWORD   <JZ,SOA,AND,A=1,B=1,D=2>                        ; 1
002742' 442010  400000

002743' 010001  010000                          MWORD   <ADDR=100,JMAP,J=101,SOA,AND,A=0,B=0,D=2>       ; 100
002744' 442000  000040
002745' 010100  000000                          MWORD   <CONT,SOA,XNOR,A=0,B=1,D=2>                     ; 101
002746' 472000  400340
002747' 010200  001777                          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=0,B=1,D=3>        ; 102
002750' 543000  640340
002751' 010301  050000                          MWORD   <CJP,J=105,SOA,OR,A=1,CENA,CCFZ>                ; 103
002752' 430410  020060
002753' 010450  000000                          MWORD   <JMAP,J=5000>                                   ; 104
002754' 000000  000040

002755' 010500  000000                          MWORD   <CONT,SOA,XNOR,A=0,B=2,D=2>                     ; 105
002756' 472001  000340
002757' 010600  001777                          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=0,B=2,D=3>        ; 106
002760' 543001  240340
002761' 010701  110000                          MWORD   <CJP,J=111,SOA,OR,A=2,CENA,CCFZ>                ; 107
002762' 430420  020060
002763' 011050  000000                          MWORD   <JMAP,J=5000>                                   ; 110
002764' 000000  000040

002765' 011100  000000                          MWORD   <CONT,SOA,XNOR,A=0,B=3,D=2>                     ; 111
002766' 472001  400340
002767' 011200  001777                          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=0,B=3,D=3>        ; 112
```

```
2073  002770' 543001  640340          MWORD   <CJP,J=115,SOA,OR,A=3,CENA,CCFZ>                ; 113
2074  002771' 011301  150000
2075  002772' 430430  020060          MWORD   <JMAP,J=5000>                                   ; 114
2076  002773' 011450  000000
2077  002774' 000000  000040
2078
2079  002775' 011500  000000          MWORD   <CONT,SOA,XNOR,A=0,B=4,D=2>                     ; 115
2080  002776' 472002  000340
2081  002777' 011600  001777          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=0,B=4,D=3>        ; 116
2082  003000' 543002  240340
2083  003001' 011701  210000          MWORD   <CJP,J=121,SOA,OR,A=4,CENA,CCFZ>                ; 117
2084  003002' 430440  020060
2085  003003' 012050  000000          MWORD   <JMAP,J=5000>                                   ; 120
2086  003004' 000000  000040
2087
2088  003005' 012100  000000          MWORD   <CONT,SOA,XNOR,A=0,B=5,D=2>                     ; 121
2089  003006' 472002  400340
2090  003007' 012200  001777          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=0,B=5,D=3>        ; 122
2091  003010' 543002  640340
2092  003011' 012301  250000          MWORD   <CJP,J=125,SOA,OR,A=5,CENA,CCFZ>                ; 123
2093  003012' 430450  020060
2094  003013' 012450  000000          MWORD   <JMAP,J=5000>                                   ; 124
2095  003014' 000000  000040
2096
2097  003015' 012500  000000          MWORD   <CONT,SOA,XNOR,A=0,B=6,D=2>                     ; 125
2098  003016' 472003  000340
2099  003017' 012600  001777          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=0,B=6,D=3>        ; 126
2100  003020' 543003  240340
2101  003021' 012701  310000          MWORD   <CJP,J=131,SOA,OR,A=6,CENA,CCFZ>                ; 127
2102  003022' 430460  020060
2103  003023' 013050  000000          MWORD   <JMAP,J=5000>                                   ; 130
2104  003024' 000000  000040
2105
2106  003025' 013100  000000          MWORD   <CONT,SOA,XNOR,A=0,B=7,D=2>                     ; 131
2107  003026' 472003  400340
2108  003027' 013200  001777          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=0,B=7,D=3>        ; 132
2109  003030' 543003  640340
2110  003031' 013301  350000          MWORD   <CJP,J=135,SOA,OR,A=7,CENA,CCFZ>                ; 133
2111  003032' 430470  020060
2112  003033' 013450  000000          MWORD   <JMAP,J=5000>                                   ; 134
2113  003034' 000000  000040
2114
2115  003035' 013500  000000          MWORD   <CONT,SOA,XNOR,A=0,B=10,D=2>                    ; 135
2116  003036' 472004  000340
2117  003037' 013600  001777          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=0,B=10,D=3>       ; 136
2118  003040' 543004  240340
2119  003041' 013701  410000          MWORD   <CJP,J=141,SOA,OR,A=10,CENA,CCFZ>               ; 137
2120  003042' 430500  020060
2121  003043' 014050  000000          MWORD   <JMAP,J=5000>                                   ; 140
2122  003044' 000000  000040
2123
2124  003045' 014100  000000          MWORD   <CONT,SOA,XNOR,A=0,B=11,D=2>                    ; 141
2125  003046' 472004  400340
2126  003047' 014200  001777          MWORD   <CONT,MGC=1777,SKCN,SDA.AND,A=0,B=11,D=3>       ; 142
2127  003050' 543004  640340
```

```
003051' 014301  450000          MWORD   <CJP,J=145,SOA,OR,A=11,CENA,CCFZ>               ; 143
003052' 430510  020060
003053' 014450  000000          MWORD   <JMAP,J=5000>                                   ; 144
003054' 000000  000040

003055' 014500  000000          MWORD   <CONT,SOA,XNOR,A=0,B=12,D=2>                    ; 145
003056' 472005  000340
003057' 014600  001777          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=0,B=12,D=3>       ; 146
003060' 543005  240340
003061' 014701  510000          MWORD   <CJP,J=151,SOA,OR,A=12,CENA,CCFZ>               ; 147
003062' 430520  020060
003063' 015050  000000          MWORD   <JMAP,J=5000>                                   ; 150
003064' 000000  000040

003065' 015100  000000          MWORD   <CONT,SOA,XNOR,A=0,B=13,D=2>                    ; 151
003066' 472005  400340
003067' 015200  001777          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=0,B=13,D=3>       ; 152
003070' 543005  640340
003071' 015301  550000          MWORD   <CJP,J=155,SOA,OR,A=13,CENA,CCFZ>               ; 153
003072' 430530  020060
003073' 015450  000000          MWORD   <JMAP,J=5000>                                   ; 154
003074' 000000  000040

003075' 015500  000000          MWORD   <CONT,SOA,XNOR,A=0,B=14,D=2>                    ; 155
003076' 472006  000340
003077' 015600  001777          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=0,B=14,D=3>       ; 156
003100' 543006  240340
003101' 015701  610000          MWORD   <CJP,J=161,SOA,OR,A=14,CENA,CCFZ>               ; 157
003102' 430540  020060
003103' 016050  000000          MWORD   <JMAP,J=5000>                                   ; 160
003104' 000000  000040

003105' 016100  000000          MWORD   <CONT,SOA,XNOR,A=0,B=15,D=2>                    ; 161
003106' 472006  400340
003107' 016200  001777          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=0,B=15,D=3>       ; 162
003110' 543006  640340
003111' 016301  650000          MWORD   <CJP,J=165,SOA,OR,A=15,CENA,CCFZ>               ; 163
003112' 430550  020060
003113' 016450  000000          MWORD   <JMAP,J=5000>                                   ; 164
003114' 000000  000040

003115' 016500  000000          MWORD   <CONT,SOA,XNOR,A=0,B=16,D=2>                    ; 165
003116' 472007  000340
003117' 016600  001777          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=0,B=16,D=3>       ; 166
003120' 543007  240340
003121' 016701  710000          MWORD   <CJP,J=171,SOA,OR,A=16,CENA,CCFZ>               ; 167
003122' 430560  020060
003123' 017050  000000          MWORD   <JMAP,J=5000>                                   ; 170
003124' 000000  000040

003125' 017100  000000          MWORD   <CONT,SOA,XNOR,A=0,B=17,D=2>                    ; 171
003126' 472007  400340
003127' 017200  001777          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=0,B=17,D=3>       ; 172
003130' 543007  640340
003131' 017301  750000          MWORD   <CJP,J=175,SOA,OR,A=17,CENA,CCFZ>               ; 173
```

```
003132' 430570  020060
003133' 017450  000000                  MWORD   <JMAP,J=5000>                                   ; 174
003134' 000000  000040

003135' 017500  000000                  MWORD   <CONT,SOA,XNOR,A=0,B=0,D=2>                     ; 175
003136' 472000  000340
003137' 017600  001777                  MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=17,B=0,D=3>       ; 176
003140' 543170  240340
003141' 017703  000000                  MWORD   <CJP,J=300,SOA,OR,A=0,CENA,CCFZ>                ; 177
003142' 430400  020060
003143' 020050  000000                  MWORD   <JMAP,J=5000>                                   ; 200
003144' 000000  000040

003145' 030000  000000                  MWORD   <ADDR=300,CONT,SOA,AND,D=2,A=16,B=16>           ; 300
003146' 442167  000340
003147' 030100  000000                  MWORD   <CONT,SOA,XNOR,A=16,B=4,D=2>                    ; 301
003150' 472162  000340
003151' 030200  001777                  MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=16,B=4,D=3>       ; 302
003152' 543162  240340
003153' 030303  050000                  MWORD   <CJP,J=305,SOA,OR,A=4,CENA,CCFZ>                ; 303
003154' 430440  020060
003155' 030450  000000                  MWORD   <JMAP,J=5000>                                   ; 304
003156' 000000  000040

003157' 030500  000000                  MWORD   <CONT,SOA,AND,A=15,B=15,D=2>                    ; 305
003160' 442156  400340
003161' 030600  000000                  MWORD   <CONT,SOA,XNOR,A=15,B=4,D=2>                    ; 306
003162' 472152  000340
003163' 030700  001777                  MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=15,B=4,D=3>       ; 307
003164' 543152  240340
003165' 031003  120000                  MWORD   <CJP,J=312,SOA,OR,A=4,CENA,CCFZ>                ; 310
003166' 430440  020060
003167' 031150  000000                  MWORD   <JMAP,J=5000>                                   ; 311
003170' 000000  000040

003171' 031200  000000                  MWORD   <CONT,SOA,AND,A=14,B=14,D=2>                    ; 312
003172' 442146  000340
003173' 031300  000000                  MWORD   <CONT,SOA,XNOR,A=14,B=4,D=2>                    ; 313
003174' 472142  000340
003175' 031400  001777                  MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=14,B=4,D=3>       ; 314
003176' 543142  240340
003177' 031503  170000                  MWORD   <CJP,J=317,SOA,OR,A=4,CENA,CCFZ>                ; 315
003200' 430440  020060
003201' 031650  000000                  MWORD   <JMAP,J=5000>                                   ; 316
003202' 000000  000040

003203' 031700  000000                  MWORD   <CONT,SOA,AND,A=13,B=13,D=2>                    ; 317
003204' 442135  400340
003205' 032000  000000                  MWORD   <CONT,SOA,XNOR,A=13,B=4,D=2>                    ; 320
003206' 472132  000340
003207' 032100  001777                  MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=13,B=4,D=3>       ; 321
003210' 543132  240340
003211' 032203  240000                  MWORD   <CJP,J=324,SOA,OR,A=4,CENA,CCFZ>                ; 322
003212' 430440  020060
003213' 032350  000000                  MWORD   <JMAP,J=5000>                                   ; 323
```

```
003214' 000000  000040

003215' 032400  000000          MWORD   <CONT,SOA,AND,A=12,B=12,D=2>                    ; 324
003216' 442125  000340
003217' 032500  000000          MWORD   <CONT,SOA,XNOR,A=12,B=4,D=2>                    ; 325
003220' 472122  000340
003221' 032600  001777          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=12,B=4,D=3>       ; 326
003222' 543122  240340
003223' 032703  310000          MWORD   <CJP,J=331,SOA,OR,A=4,CENA,CCFZ>                ; 327
003224' 430440  020060
003225' 033050  000000          MWORD   <JMAP,J=5000>                                   ; 330
003226' 000000  000040

003227' 033100  000000          MWORD   <CONT,SOA,AND,A=11,B=11,D=2>                    ; 331
003230' 442114  400340
003231' 033200  000000          MWORD   <CONT,SOA,XNOR,A=11,B=4,D=2>                    ; 332
003232' 472112  000340
003233' 033300  001777          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=11,B=4,D=3>       ; 333
003234' 543112  240340
003235' 033403  360000          MWORD   <CJP,J=336,SOA,OR,A=4,CENA,CCFZ>                ; 334
003236' 430440  020060
003237' 033550  000000          MWORD   <JMAP,J=5000>                                   ; 335
003240' 000000  000040

003241' 033600  000000          MWORD   <CONT,SOA,AND,A=10,B=10,D=2>                    ; 336
003242' 442104  000340
003243' 033700  000000          MWORD   <CONT,SOA,XNOR,A=10,B=4,D=2>                    ; 337
003244' 472102  000340
003245' 034000  001777          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=10,B=4,D=3>       ; 340
003246' 543102  240340
003247' 034103  430000          MWORD   <CJP,J=343,SOA,OR,A=4,CENA,CCFZ>                ; 341
003250' 430440  020060
003251' 034250  000000          MWORD   <JMAP,J=5000>                                   ; 342
003252' 000000  000040

003253' 034300  000000          MWORD   <CONT,SOA,AND,A=7,B=7,D=2>                      ; 343
003254' 442073  400340
003255' 034400  000000          MWORD   <CONT,SOA,XNOR,A=7,B=4,D=2>                     ; 344
003256' 472072  000340
003257' 034500  001777          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=7,B=4,D=3>        ; 345
003260' 543072  240340
003261' 034603  500000          MWORD   <CJP,J=350,SOA,OR,A=4,CENA,CCFZ>                ; 346
003262' 430440  020060
003263' 034750  000000          MWORD   <JMAP,J=5000>                                   ; 347
003264' 000000  000040

003265' 035000  000000          MWORD   <CONT,SOA,AND,A=6,B=6,D=2>                      ; 350
003266' 442063  000340
003267' 035100  000000          MWORD   <CONT,SOA,XNOR,A=6,B=4,D=2>                     ; 351
003270' 472062  000340
003271' 035200  001777          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=6,B=4,D=3>        ; 352
003272' 543062  240340
003273' 035303  550000          MWORD   <CJP,J=355,SOA,OR,A=4,CENA,CCFZ>                ; 353
003274' 430440  020060
003275' 035450  000000          MWORD   <JMAP,J=5000>                                   ; 354
```

```
003276' 000000  000040

003277' 035500  000000              MWORD   <CONT,SOA,AND,A=5,B=5,D=2>                          ; 355
003300' 442052  400340
003301' 035600  000000              MWORD   <CONT,SOA,XNOR,A=5,B=4,D=2>                         ; 356
003302' 472052  000340
003303' 035700  001777              MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=5,B=4,D=3>            ; 357
003304' 543052  240340
003305' 036003  620000              MWORD   <CJP,J=362,SOA,OR,A=4,CENA,CCFZ>                    ; 360
003306' 430440  020060
003307' 036150  000000              MWORD   <JMAP,J=5000>                                       ; 361
003310' 000000  000040

003311' 036200  000000              MWORD   <CONT,SOA,AND,A=3,B=3,D=2>                          ; 362
003312' 442031  400340
003313' 036300  000000              MWORD   <CONT,SOA,XNOR,A=3,B=4,D=2>                         ; 363
003314' 472032  000340
003315' 036400  001777              MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=3,B=4,D=3>            ; 364
003316' 543032  240340
003317' 036503  670000              MWORD   <CJP,J=367,SOA,OR,A=4,CENA,CCFZ>                    ; 365
003320' 430440  020060
003321' 036650  000000              MWORD   <JMAP,J=5000>                                       ; 366
003322' 000000  000040

003323' 036700  000000              MWORD   <CONT,SOA,AND,A=2,B=2,D=2>                          ; 367
003324' 442021  000340
003325' 037000  000000              MWORD   <CONT,SOA,XNOR,A=2,B=4,D=2>                         ; 370
003326' 472022  000340
003327' 037100  001777              MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=2,B=4,D=3>            ; 371
003330' 543022  240340
003331' 037203  740000              MWORD   <CJP,J=374,SOA,OR,A=4,CENA,CCFZ>                    ; 372
003332' 430440  020060
003333' 037350  000000              MWORD   <JMAP,J=5000>                                       ; 373
003334' 000000  000040

003335' 037400  000000              MWORD   <CONT,SOA,AND,A=1,B=1,D=2>                          ; 374
003336' 442010  400340
003337' 037500  000000              MWORD   <CONT,SOA,XNOR,A=1,B=4,D=2>                         ; 375
003340' 472012  000340
003341' 037600  001777              MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=1,B=4,D=3>            ; 376
003342' 543012  240340
003343' 037704  010000              MWORD   <CJP,J=401,SOA,OR,A=4,CENA,CCFZ>                    ; 377
003344' 430440  020060
003345' 040050  000000              MWORD   <JMAP,J=5000>                                       ; 400
003346' 000000  000040

003347' 040100  000000              MWORD   <CONT,SOA,AND,A=16,B=16,D=2>                        ; 401
003350' 442167  000340
003351' 040200  000000              MWORD   <CONT,SOA,XNOR,A=16,B=5,D=2>                        ; 402
003352' 472162  400340
003353' 040300  001777              MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=16,B=5,D=3>           ; 403
003354' 543162  640340
003355' 040404  060000              MWORD   <CJP,J=406,SOA,OR,A=5,CENA,CCFZ>                    ; 404
003356' 430450  020060
003357' 040550  000000              MWORD   <JMAP,J=5000>                                       ; 405
```

```
003360' 000000  000040

003361' 040600  000000                          MWORD   <CONT,SOA,AND,A=16,B=16,D=2>                    ; 406
003362' 442167  000340
003363' 040700  000000                          MWORD   <CONT,SOA,XNOR,A=16,B=6,D=2>                    ; 407
003364' 472163  000340
003365' 041000  001777                          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=16,B=6,D=3>       ; 410
003366' 543163  240340
003367' 041104  130000                          MWORD   <CJP,J=413,SOA,OR,A=6,CENA,CCFZ>                ; 411
003370' 430460  020060
003371' 041250  000000                          MWORD   <JMAP,J=5000>                                   ; 412
003372' 000000  000040

003373' 041300  000000                          MWORD   <CONT,SOA,AND,A=16,B=16,D=2>                    ; 413
003374' 442167  000340
003375' 041400  000000                          MWORD   <CONT,SOA,XNOR,A=16,B=7,D=2>                    ; 414
003376' 472163  400340
003377' 041500  001777                          MWORD   <CONT,MGC=1777,SKCN,SDA,AND,A=16,B=7,D=3>       ; 415
003400' 543163  640340
003401' 041640  000000                          MWORD   <CJP,J=4000,SOA,OR,A=7,CENA,CCFZ>               ; 416
003402' 430470  020060
003403' 041750  000000                          MWORD   <JMAP,J=5000>                                   ; 417
003404' 000000  000040

                                        ;       Correct address

003405' 400040  000000                          MWORD   <ADDR=4000,JMAP,J=4000,BAD>                     ; 4000
003406' 000000  000041

                                        ;       Error address

003407' 500050  000000                          MWORD   <ADDR=5000,JMAP,J=5000,BAD>                     ; 5000
003410' 000000  000041
003411' 777777  777777                          -1

                                        ;#**********************************************************************
                                        ; End of 2901 Tests (Part 4)
                                        ;#**********************************************************************

                                                XLIST

NO ERRORS DETECTED

PROGRAM BREAK IS 003676
CPU TIME USED 04:47.686

175P CORE USED
```

```
DFPTA7  MACRO %53B(1242) 09:06  1-Jul-85 Page S-1
DFPTA7  MAC       18-Apr-85 10:59          SYMBOL TABLE

AAPNT           000000  ext   TA37            002034'       TX40            002301'       .MJ             007777  spd
AEXEC           002727' ext   TA40            002261'       TX41            002523'       .MJMAP          000017  spd
ALEBF           002533' ext   TA41            002503'       TX42            002734'       .MJZ            000017  spd
ALFLS           002512' ext   TA42            002723'       TXALL   060000  000000  spd   .MLDCT          000017  spd
ALU     020000  000000  spd   TG33            000013'       TXNOT   040000  000000  spd   .MMGC           001777  spd
BBPNT           000000  ext   TG34            000147'       TXTINH          000200  spd   .MNAND          000007  spd
BEXEC           000321' ext   TG35            000310'       Z7              000000'       .MOENA          000001  spd
CALL    200000  000000  spd   TG36            001343'       ZALU    000002  000000  spd   .MOR            000007  spd
E23             000027  spd   TG37            002025'       $ARG2           000001        .MPLUS          000007  spd
EBUS    400000  000000  spd   TG40            002252'       $B              000044        .MRPCT          000017  spd
ERFLG           000015        TG41            002474'       $CHR            424144        .MSOA           000007  spd
GO      260740  000000        TG42            002714'       $GARG           000000        .MSOB           000007  spd
IPACLR          002722' ext   TL33            000020'       %ADDR           005001  spd   .MSOQ           000007  spd
LAST    010000  000000  spd   TL34            000154'       %ML     500050  000000  spd   .MSDO           000007  spd
MA33            000102'       TL35            000315'       %MR             000041  spd   .MSDA           000007  spd
MA34            000176'       TL36            001350'       .LA             000000  spd   .MSELE          000007  spd
MA34PN          000200'       TL37            002032'       .LADDR  000100  000000  spd   .MSKCN          000037  spd
MA35            000523'       TL40            002257'       .LAND           000000  spd   .MXNOR          000007  spd
MA35PN          000525'       TL41            002501'       .LB             000000  spd   .MXOR           000007  spd
MA36            001373'       TL42            002721'       .LBAD           000000  spd   .RA     000010  000000  spd
MA36PN          001375'       TLOAD           002717' ext   .LCCFZ          000000  spd   .RAND   04000C  000000  spd
MA37            002055'       TRACE           002715' ext   .LCENA          000000  spd   .RB             400000  spd
MA37PN          002057'       TS33            000035'       .LCJP           000000  spd   .RBAD           000001  spd
MA40            002302'       TS33A0          000062'       .LCONT          000000  spd   .RCCFZ          020000  spd
MA40PN          002304'       TS33A1          000070'       .LCRY           000000  spd   .RCENA  000400  000000  spd
MA41            002524'       TS33AD          000052'       .LD             000000  spd   .RCJP           000060  spd
MA41PN          002526'       TS33IN          000045'       .LJ             010000  spd   .RCONT          000340  spd
MA42            002735'       TS33L           000040'       .LJMAP          000000  spd   .RCRY           000400  spd
MLAST   400000  000000  spd   TS33PT          003671'       .LJZ            000000  spd   .RD     001000  000000  spd
MSG     100000  000000  spd   TS33ST          003672'       .LLDCT          000000  spd   .RJ             000000  spd
NDMP    000400  000000  spd   TS35            000332'       .LMGC           000001  spd   .RJMAP          000040  spd
P               000017        TS35A0          000360'       .LNAND          000000  spd   .RJZ            000000  spd
PNTHW   037540  000000        TS35A1          000364'       .LOENA          002000  spd   .RLDCT          000300  spd
PNTMSG  037000  000000        TS35AD          000345'       .LOR            000000  spd   .RMGC           000000  spd
PNTOCS  037700  000003        TS35IN          000340'       .LPLUS          000000  spd   .RNAND  050000  000000  spd
RTN     263740  000000        TS35L           000333'       .LRPCT          000000  spd   .ROENA          000000  spd
SCOPER  027000  000000        TS35PA          003673'       .LSOA           000000  spd   .ROR    030000  000000  spd
SCOSW           002526' ext   TS35RG          003674'       .LSOB           000000  spd   .RPLUS          000000  spd
SSCALL          000003  spd   TS35ST          003675'       .LSOQ           000000  spd   .RRPCT          000220  spd
SSCHK           000004  spd   TS35T1          000401'       .LSDO           000000  spd   .RSOA   400000  000000  spd
SSJRST          000005  spd   TS35T2          000405'       .LSDA           000000  spd   .RSOB   300000  000000  spd
SSLAST          000000  spd   TSTA33          000000' ent   .LSELE          000000  spd   .RSOQ   200000  000000  spd
SSSTRT          000001  spd   TSTA34          000135' ent   .LSKCN          000000  spd   .RSDO   700000  000000  spd
T33M            000104'       TSTA35          000277' ent   .LXNOR          000000  spd   .RSDA   500000  000000  spd
T34M            000210'       TSTA36          001333' ent   .LXOR           000000  spd   .RSELE          005000  spd
T35M            000532'       TSTA37          002016' ent   .MA             000017  spd   .RSKCN          240000  spd
T36M            001407'       TSTA40          002244' ent   .MAND           000007  spd   .RXNOR  070000  000000  spd
T37M            002071'       TSTA41          002467' ent   .MB             000017  spd   .RXOR   060000  000000  spd
T40M            002314'       TSTA42          002707' ent   .MBAD           000001  spd
T41M            002536'       TSTSUB          000317' ext   .MCCFZ          000037  spd
T42M            002737'       TX33            000034'       .MCENA          000001  spd
TA33            000024'       TX34            000175'       .MCJP           000017  spd
TA34            000156'       TX35            000331'       .MCONT          000017  spd
TA35            000321'       TX36            001372'       .MCRY           000001  spd
TA36            001352'       TX37            002054'       .MD             000007  spd
```

```
AAPNT      14#   2041
AEXEC      14#    251   1075   1433   1634   1832   2025
ALEBF      14#    256    282   1080   1107   1438   1465   1639   1666   1837   1864
ALFLS      14#    254   1078   1436   1637   1835
ALU        73     222    401   1048   1407   1609   1808   2001
BBPNT      14#    166    581
BEXEC      14#    101    427
CALL      166     273    580    581   1098   1456   1657   1855   2041
E23        75     224    403   1050   1409   1611   1810   2003
EBUS       73     222    401   1048   1407   1609   1808   2001
ERFLG      94     104    111    242    253    255    258    420    430    437   1066   1077   1079   1083
         1424    1435   1437   1441   1625   1636   1638   1642   1823   1834   1836   1840   2016   2026
IPACLR     18#     95    243    421   1067   1425   1626   1824   2017
LAST      166     273    581   1098   1456   1657   1855   2041
MA33      108     165#
MA34      262     272#
MA34PN    273     275#
MA35      434     580#
MA35PN    580     583#
MA36     1087    1097#
MA36PN   1098    1100#
MA37     1445    1455#
MA37PN   1456    1458#
MA40     1646    1656#
MA40PN   1657    1659#
MA41     1844    1854#
MA41PN   1855    1857#
MA42     2030    2040#
MLAST      75     224    403   1050   1409   1611   1810   2003
MSG       165     272   1097   1455   1656   1854   2040
NDMP       73     222    401   1048   1407   1609   1808   2001
P         160     487
SCOSW      22#    275   1100   1458   1659   1857
SSCALL    121     122    445    446
SSCHK     122     123    446    447
SSJRST    125     126    449    450
SSLAST    126     127    450    451
SSSTRT    119     120    121    123    124    125    447    448    449
T33M       74      88    170#
T34M      223     236    288#
T35M      402     414    593#
T36M     1049    1060   1115#
T37M     1408    1418   1473#
T40M     1610    1619   1672#
T41M     1809    1817   1870#
T42M     2002    2010   2045#
TA33      101#    103    111
TA34      247#
TA35      427#    429    437
TA36     1071#
TA37     1429#
TA40     1630#
TA41     1828#
```

```
TA42       2021#
TG33         72      86#
TG34        221     234#
TG35        400     412#
TG36       1047    1058#
TG37       1406    1416#
TG40       1608    1617#
TG41       1807    1815#
TG42       2000    2008#
TL33         94#    109
TL34        242#    263
TL35        420#    435
TL36       1066#   1088
TL37       1424#   1446
TL40       1625#   1647
TL41       1823#   1845
TL42       2016#   2031
TLOAD        14#     89     237     415     1061    1419    1620    1818    2011
TRACE        14#     87     235     413     1059    1417    1618    1816    2009
TS33         97     119#
TS33A0      138     145#
TS33A1      143     148     152#
TS33AD      122     136#
TS33IN      121     130#
TS33L       122#    125
TS33PT      131#    131     136     152#     152
TS33ST      133#    133     137     153#     153
TS35        423     445#
TS35A0      464     472#
TS35A1      470     473     476#
TS35AD      446     460#
TS35IN      445     454#
TS35L       446#    449
TS35PA      457#    457     461     469      475     484
TS35RG      454#    454     465     477      584
TS35ST      455#    455     460     468      474
TS35T1      479     492#
TS35T2      462     485     499#
TSTA33       10      72#
TSTA34       10      76     221#
TSTA35       10      77     225     400#
TSTA36       10      78     226     404     1047#
TSTA37       10      79     227     405     1051    1406#
TSTA40       10      80     228     406     1052    1410    1608#
TSTA41       10      81     229     407     1053    1411    1612    1807#
TSTA42       10    2000#
TSTSUB       14#     96     422
TX33        102     110     115#
TX34        264     268#
TX35        428     436     441#
TX36       1089    1093#
TX37       1447    1451#
TX40       1648    1652#
```

```
TX41     1846     1850#
TX42     2032     2036#
TXALL     166      273      581     1098     1456     1657     1854     1855     2040     2041
TXNOT     165      272      580     1097     1455     1656
TXTINH    276      279     1101     1104     1109     1459     1462     1467     1660     1663     1858     1861
Z7         29#      86      234      412     1058     1416     1617     1815     2008
ZALU       73      222      401     1048     1407     1609     1808     2001
$ARG2     170#     170      172#     172      174#     174      177#     177      179#     179      181#     181      184#     184
          186#     186      188#     188      191#     191      193#     193      195#     195      288#     288      290#     290
          295#     295      297#     297      299#     299      301#     301      303#     303      305#     305      310#     310
          312#     312      314#     314      316#     316      318#     318      320#     320      325#     325      327#     327
          329#     329      331#     331      333#     333      335#     335      340#     340      342#     342      344#     344
          346#     346      348#     348      350#     350      353#     353      593#     593      595#     595      597#     597
          599#     599      601#     601      603#     603      605#     605      607#     607      609#     609      611#     611
          613#     613      615#     615      620#     620      622#     622      624#     624      626#     626      628#     628
          630#     630      632#     632      634#     634      636#     636      638#     638      640#     640      642#     642
          647#     647      649#     649      651#     651      653#     653      655#     655      657#     657      659#     659
          661#     661      663#     663      665#     665      667#     667      669#     669      674#     674      676#     676
          678#     678      680#     680      682#     682      684#     684      686#     686      688#     688      690#     690
          692#     692      694#     694      696#     696      701#     701      703#     703      705#     705      707#     707
          709#     709      711#     711      713#     713      715#     715      717#     717      719#     719      721#     721
          723#     723      728#     728      730#     730      732#     732      734#     734      736#     736      738#     738
          740#     740      742#     742      744#     744      746#     746      748#     748      750#     750      755#     755
          757#     757      759#     759      761#     761      763#     763      765#     765      767#     767      769#     769
          771#     771      773#     773      775#     775      777#     777      782#     782      784#     784      786#     786
          788#     788      790#     790      792#     792      794#     794      796#     796      798#     798      800#     800
          802#     802      804#     804      809#     809      811#     811      813#     813      815#     815      817#     817
          819#     819      821#     821      823#     823      825#     825      827#     827      829#     829      831#     831
          836#     836      838#     838      840#     840      842#     842      844#     844      846#     846      848#     848
          850#     850      852#     852      854#     854      856#     856      858#     858      863#     863      865#     865
          867#     867      869#     869      871#     871      873#     873      875#     875      877#     877      879#     879
          881#     881      883#     883      885#     885      890#     890      892#     892      894#     894      896#     896
          898#     898      900#     900      902#     902      904#     904      906#     906      908#     908      910#     910
          912#     912      917#     917      919#     919      921#     921      923#     923      925#     925      927#     927
          929#     929      931#     931      933#     933      935#     935      937#     937      939#     939      944#     944
          946#     946      948#     948      950#     950      952#     952      954#     954      956#     956      958#     958
          960#     960      962#     962      964#     964      966#     966      971#     971      973#     973      975#     975
          977#     977      979#     979      981#     981      983#     983      985#     985      987#     987      989#     989
          991#     991      993#     993      998#     998     1000#    1000     1002#    1002     1004#    1004     1006#    1006
         1008#    1008     1010#    1010     1012#    1012     1014#    1014     1016#    1016     1018#    1018     1020#    1020
         1115#    1115     1117#    1117     1119#    1119     1121#    1121     1123#    1123     1125#    1125     1127#    1127
         1129#    1129     1131#    1131     1133#    1133     1135#    1135     1137#    1137     1139#    1139     1141#    1141
         1143#    1143     1145#    1145     1147#    1147     1149#    1149     1151#    1151     1153#    1153     1155#    1155
         1157#    1157     1159#    1159     1161#    1161     1163#    1163     1165#    1165     1167#    1167     1169#    1169
         1171#    1171     1173#    1173     1175#    1175     1177#    1177     1179#    1179     1182#    1182     1184#    1184
         1186#    1186     1188#    1188     1190#    1190     1192#    1192     1194#    1194     1196#    1196     1198#    1198
         1200#    1200     1202#    1202     1204#    1204     1206#    1206     1208#    1208     1210#    1210     1212#    1212
         1214#    1214     1216#    1216     1218#    1218     1220#    1220     1222#    1222     1224#    1224     1226#    1226
         1228#    1228     1230#    1230     1232#    1232     1234#    1234     1236#    1236     1238#    1238     1240#    1240
         1242#    1242     1244#    1244     1247#    1247     1249#    1249     1251#    1251     1253#    1253     1255#    1255
         1257#    1257     1259#    1259     1261#    1261     1263#    1263     1265#    1265     1267#    1267     1269#    1269
         1271#    1271     1273#    1273     1275#    1275     1277#    1277     1279#    1279     1281#    1281     1283#    1283
         1285#    1285     1287#    1287     1289#    1289     1291#    1291     1293#    1293     1295#    1295     1297#    1297
```

```
        1299#  1299   1301#  1301   1303#  1303   1305#  1305   1307#  1307   1309#  1309   1312#  1312     SEQ 1353
        1314#  1314   1316#  1316   1318#  1318   1320#  1320   1322#  1322   1324#  1324   1326#  1326
        1328#  1328   1330#  1330   1332#  1332   1334#  1334   1336#  1336   1338#  1338   1340#  1340
        1342#  1342   1344#  1344   1346#  1346   1348#  1348   1350#  1350   1352#  1352   1354#  1354
        1356#  1356   1358#  1358   1360#  1360   1362#  1362   1364#  1364   1366#  1366   1368#  1368
        1370#  1370   1372#  1372   1374#  1374   1377#  1377   1379#  1379   1473#  1473   1475#  1475
        1477#  1477   1479#  1479   1481#  1481   1483#  1483   1485#  1485   1487#  1487   1489#  1489
        1491#  1491   1493#  1493   1495#  1495   1497#  1497   1499#  1499   1501#  1501   1503#  1503
        1505#  1505   1507#  1507   1509#  1509   1511#  1511   1513#  1513   1515#  1515   1517#  1517
        1519#  1519   1521#  1521   1523#  1523   1525#  1525   1527#  1527   1529#  1529   1531#  1531
        1533#  1533   1535#  1535   1537#  1537   1539#  1539   1541#  1541   1543#  1543   1545#  1545
        1547#  1547   1549#  1549   1551#  1551   1553#  1553   1555#  1555   1557#  1557   1559#  1559
        1561#  1561   1563#  1563   1565#  1565   1567#  1567   1569#  1569   1571#  1571   1573#  1573
        1576#  1576   1578#  1578   1672#  1672   1674#  1674   1676#  1676   1678#  1678   1680#  1680
        1682#  1682   1684#  1684   1686#  1686   1688#  1688   1690#  1690   1692#  1692   1694#  1694
        1696#  1696   1698#  1698   1700#  1700   1702#  1702   1704#  1704   1706#  1706   1708#  1708
        1710#  1710   1712#  1712   1714#  1714   1716#  1716   1718#  1718   1720#  1720   1722#  1722
        1724#  1724   1726#  1726   1728#  1728   1730#  1730   1732#  1732   1734#  1734   1736#  1736
        1738#  1738   1740#  1740   1742#  1742   1744#  1744   1746#  1746   1748#  1748   1750#  1750
        1752#  1752   1754#  1754   1756#  1756   1758#  1758   1760#  1760   1762#  1762   1764#  1764
        1766#  1766   1768#  1768   1770#  1770   1772#  1772   1775#  1775   1777#  1777   1870#  1870
        1872#  1872   1874#  1874   1876#  1876   1878#  1878   1880#  1880   1882#  1882   1884#  1884
        1886#  1886   1888#  1888   1890#  1890   1892#  1892   1894#  1894   1896#  1896   1898#  1898
        1900#  1900   1902#  1902   1904#  1904   1906#  1906   1908#  1908   1910#  1910   1912#  1912
        1914#  1914   1916#  1916   1918#  1918   1920#  1920   1922#  1922   1924#  1924   1926#  1926
        1928#  1928   1930#  1930   1932#  1932   1934#  1934   1936#  1936   1938#  1938   1940#  1940
        1942#  1942   1944#  1944   1946#  1946   1948#  1948   1950#  1950   1952#  1952   1954#  1954
        1956#  1956   1958#  1958   1960#  1960   1962#  1962   1964#  1964   1966#  1966   1968#  1968
        1970#  1970   1973#  1973   2045#  2045   2047#  2047   2050#  2050   2052#  2052   2054#  2054
        2056#  2056   2058#  2058   2061#  2061   2063#  2063   2065#  2065   2067#  2067   2070#  2070
        2072#  2072   2074#  2074   2076#  2076   2079#  2079   2081#  2081   2083#  2083   2085#  2085
        2088#  2088   2090#  2090   2092#  2092   2094#  2094   2097#  2097   2099#  2099   2101#  2101
        2103#  2103   2106#  2106   2108#  2108   2110#  2110   2112#  2112   2115#  2115   2117#  2117
        2119#  2119   2121#  2121   2124#  2124   2126#  2126   2128#  2128   2130#  2130   2133#  2133
        2135#  2135   2137#  2137   2139#  2139   2142#  2142   2144#  2144   2146#  2146   2148#  2148
        2151#  2151   2153#  2153   2155#  2155   2157#  2157   2160#  2160   2162#  2162   2164#  2164
        2166#  2166   2169#  2169   2171#  2171   2173#  2173   2175#  2175   2178#  2178   2180#  2180
        2182#  2182   2184#  2184   2187#  2187   2189#  2189   2191#  2191   2193#  2193   2196#  2196
        2198#  2198   2200#  2200   2202#  2202   2204#  2204   2207#  2207   2209#  2209   2211#  2211
        2213#  2213   2215#  2215   2218#  2218   2220#  2220   2222#  2222   2224#  2224   2226#  2226
        2229#  2229   2231#  2231   2233#  2233   2235#  2235   2237#  2237   2240#  2240   2242#  2242
        2244#  2244   2246#  2246   2248#  2248   2251#  2251   2253#  2253   2255#  2255   2257#  2257
        2259#  2259   2262#  2262   2264#  2264   2266#  2266   2268#  2268   2270#  2270   2273#  2273
        2275#  2275   2277#  2277   2279#  2279   2281#  2281   2284#  2284   2286#  2286   2288#  2288
        2290#  2290   2292#  2292   2295#  2295   2297#  2297   2299#  2299   2301#  2301   2303#  2303
        2306#  2306   2308#  2308   2310#  2310   2312#  2312   2314#  2314   2317#  2317   2319#  2319
        2321#  2321   2323#  2323   2325#  2325   2328#  2328   2330#  2330   2332#  2332   2334#  2334
        2336#  2336   2339#  2339   2341#  2341   2343#  2343   2345#  2345   2347#  2347   2350#  2350
        2352#  2352   2354#  2354   2356#  2356   2358#  2358   2361#  2361   2363#  2363   2365#  2365
        2367#  2367   2369#  2369   2374#  2374   2379#  2379
$B       170#   170    172#   172    174#   174    177#   177    179#   179    181#   181    184#   184
         186#   186    188#   188    191#   191    193#   193    195#   195    288#   288    290#   290
         295#   295    297#   297    299#   299    301#   301    303#   303    305#   305    310#   310
```

```
         312#   312    314#   314    316#   316    318#   318    320#   320    325#   325    327#   327
```

1341

```
312#   312   314#   314   316#   316   318#   318   320#   320   325#   325   327#   327     SEQ 1354
329#   329   331#   331   333#   333   335#   335   340#   340   342#   342   344#   344
346#   346   348#   348   350#   350   353#   353   593#   593   595#   595   597#   597
599#   599   601#   601   603#   603   605#   605   607#   607   609#   609   611#   611
613#   613   615#   615   620#   620   622#   622   624#   624   626#   626   628#   628
630#   630   632#   632   634#   634   636#   636   638#   638   640#   640   642#   642
647#   647   649#   649   651#   651   653#   653   655#   655   657#   657   659#   659
661#   661   663#   663   665#   665   667#   667   669#   669   674#   674   676#   676
678#   678   680#   680   682#   682   684#   684   686#   686   688#   688   690#   690
692#   692   694#   694   696#   696   701#   701   703#   703   705#   705   707#   707
709#   709   711#   711   713#   713   715#   715   717#   717   719#   719   721#   721
723#   723   728#   728   730#   730   732#   732   734#   734   736#   736   738#   738
740#   740   742#   742   744#   744   746#   746   748#   748   750#   750   755#   755
757#   757   759#   759   761#   761   763#   763   765#   765   767#   767   769#   769
771#   771   773#   773   775#   775   777#   777   782#   782   784#   784   786#   786
788#   788   790#   790   792#   792   794#   794   796#   796   798#   798   800#   800
802#   802   804#   804   809#   809   811#   811   813#   813   815#   815   817#   817
819#   819   821#   821   823#   823   825#   825   827#   827   829#   829   831#   831
836#   836   838#   838   840#   840   842#   842   844#   844   846#   846   848#   848
850#   850   852#   852   854#   854   856#   856   858#   858   863#   863   865#   865
867#   867   869#   869   871#   871   873#   873   875#   875   877#   877   879#   879
881#   881   883#   883   885#   885   890#   890   892#   892   894#   894   896#   896
898#   898   900#   900   902#   902   904#   904   906#   906   908#   908   910#   910
912#   912   917#   917   919#   919   921#   921   923#   923   925#   925   927#   927
929#   929   931#   931   933#   933   935#   935   937#   937   939#   939   944#   944
946#   946   948#   948   950#   950   952#   952   954#   954   956#   956   958#   958
960#   960   962#   962   964#   964   966#   966   971#   971   973#   973   975#   975
977#   977   979#   979   981#   981   983#   983   985#   985   987#   987   989#   989
991#   991   993#   993   998#   998  1000#  1000  1002#  1002  1004#  1004  1006#  1006
1008#  1008  1010#  1010  1012#  1012  1014#  1014  1016#  1016  1018#  1018  1020#  1020
1115#  1115  1117#  1117  1119#  1119  1121#  1121  1123#  1123  1125#  1125  1127#  1127
1129#  1129  1131#  1131  1133#  1133  1135#  1135  1137#  1137  1139#  1139  1141#  1141
1143#  1143  1145#  1145  1147#  1147  1149#  1149  1151#  1151  1153#  1153  1155#  1155
1157#  1157  1159#  1159  1161#  1161  1163#  1163  1165#  1165  1167#  1167  1169#  1169
1171#  1171  1173#  1173  1175#  1175  1177#  1177  1179#  1179  1182#  1182  1184#  1184
1186#  1186  1188#  1188  1190#  1190  1192#  1192  1194#  1194  1196#  1196  1198#  1198
1200#  1200  1202#  1202  1204#  1204  1206#  1206  1208#  1208  1210#  1210  1212#  1212
1214#  1214  1216#  1216  1218#  1218  1220#  1220  1222#  1222  1224#  1224  1226#  1226
1228#  1228  1230#  1230  1232#  1232  1234#  1234  1236#  1236  1238#  1238  1240#  1240
1242#  1242  1244#  1244  1247#  1247  1249#  1249  1251#  1251  1253#  1253  1255#  1255
1257#  1257  1259#  1259  1261#  1261  1263#  1263  1265#  1265  1267#  1267  1269#  1269
1271#  1271  1273#  1273  1275#  1275  1277#  1277  1279#  1279  1281#  1281  1283#  1283
1285#  1285  1287#  1287  1289#  1289  1291#  1291  1293#  1293  1295#  1295  1297#  1297
1299#  1299  1301#  1301  1303#  1303  1305#  1305  1307#  1307  1309#  1309  1312#  1312
1314#  1314  1316#  1316  1318#  1318  1320#  1320  1322#  1322  1324#  1324  1326#  1326
1328#  1328  1330#  1330  1332#  1332  1334#  1334  1336#  1336  1338#  1338  1340#  1340
1342#  1342  1344#  1344  1346#  1346  1348#  1348  1350#  1350  1352#  1352  1354#  1354
1356#  1356  1358#  1358  1360#  1360  1362#  1362  1364#  1364  1366#  1366  1368#  1368
1370#  1370  1372#  1372  1374#  1374  1377#  1377  1379#  1379  1473#  1473  1475#  1475
1477#  1477  1479#  1479  1481#  1481  1483#  1483  1485#  1485  1487#  1487  1489#  1489
1491#  1491  1493#  1493  1495#  1495  1497#  1497  1499#  1499  1501#  1501  1503#  1503
1505#  1505  1507#  1507  1509#  1509  1511#  1511  1513#  1513  1515#  1515  1517#  1517
1519#  1519  1521#  1521  1523#  1523  1525#  1525  1527#  1527  1529#  1529  1531#  1531
```

.LJ

```
1533#  1533  1535#  1535  1537#  1537  1539#  1539  1541#  1541  1543#  1543  1545#  1545
```

```
          1533#  1533    1535#  1535    1537#  1537    1539#  1539    1541#  1541    1543#  1543    1545#  1545     SEQ 1355
          1547#  1547    1549#  1549    1551#  1551    1553#  1553    1555#  1555    1557#  1557    1559#  1559
          1561#  1561    1563#  1563    1565#  1565    1567#  1567    1569#  1569    1571#  1571    1573#  1573
          1576#  1576    1578#  1578    1672#  1672    1674#  1674    1676#  1676    1678#  1678    1680#  1680
          1682#  1682    1684#  1684    1686#  1686    1688#  1688    1690#  1690    1692#  1692    1694#  1694
          1696#  1696    1698#  1698    1700#  1700    1702#  1702    1704#  1704    1706#  1706    1708#  1708
          1710#  1710    1712#  1712    1714#  1714    1716#  1716    1718#  1718    1720#  1720    1722#  1722
          1724#  1724    1726#  1726    1728#  1728    1730#  1730    1732#  1732    1734#  1734    1736#  1736
          1738#  1738    1740#  1740    1742#  1742    1744#  1744    1746#  1746    1748#  1748    1750#  1750
          1752#  1752    1754#  1754    1756#  1756    1758#  1758    1760#  1760    1762#  1762    1764#  1764
          1766#  1766    1768#  1768    1770#  1770    1772#  1772    1775#  1775    1777#  1777    1870#  1870
          1872#  1872    1874#  1874    1876#  1876    1878#  1878    1880#  1880    1882#  1882    1884#  1884
          1886#  1886    1888#  1888    1890#  1890    1892#  1892    1894#  1894    1896#  1896    1898#  1898
          1900#  1900    1902#  1902    1904#  1904    1906#  1906    1908#  1908    1910#  1910    1912#  1912
          1914#  1914    1916#  1916    1918#  1918    1920#  1920    1922#  1922    1924#  1924    1926#  1926
          1928#  1928    1930#  1930    1932#  1932    1934#  1934    1936#  1936    1938#  1938    1940#  1940
          1942#  1942    1944#  1944    1946#  1946    1948#  1948    1950#  1950    1952#  1952    1954#  1954
          1956#  1956    1958#  1958    1960#  1960    1962#  1962    1964#  1964    1966#  1966    1968#  1968
          1970#  1970    1973#  1973    2045#  2045    2047#  2047    2050#  2050    2052#  2052    2054#  2054
          2056#  2056    2058#  2058    2061#  2061    2063#  2063    2065#  2065    2067#  2067    2070#  2070
          2072#  2072    2074#  2074    2076#  2076    2079#  2079    2081#  2081    2083#  2083    2085#  2085
          2088#  2088    2090#  2090    2092#  2092    2094#  2094    2097#  2097    2099#  2099    2101#  2101
          2103#  2103    2106#  2106    2108#  2108    2110#  2110    2112#  2112    2115#  2115    2117#  2117
          2119#  2119    2121#  2121    2124#  2124    2126#  2126    2128#  2128    2130#  2130    2133#  2133
          2135#  2135    2137#  2137    2139#  2139    2142#  2142    2144#  2144    2146#  2146    2148#  2148
          2151#  2151    2153#  2153    2155#  2155    2157#  2157    2160#  2160    2162#  2162    2164#  2164
          2166#  2166    2169#  2169    2171#  2171    2173#  2173    2175#  2175    2178#  2178    2180#  2180
          2182#  2182    2184#  2184    2187#  2187    2189#  2189    2191#  2191    2193#  2193    2196#  2196
          2198#  2198    2200#  2200    2202#  2202    2204#  2204    2207#  2207    2209#  2209    2211#  2211
          2213#  2213    2215#  2215    2218#  2218    2220#  2220    2222#  2222    2224#  2224    2226#  2226
          2229#  2229    2231#  2231    2233#  2233    2235#  2235    2237#  2237    2240#  2240    2242#  2242
          2244#  2244    2246#  2246    2248#  2248    2251#  2251    2253#  2253    2255#  2255    2257#  2257
          2259#  2259    2262#  2262    2264#  2264    2266#  2266    2268#  2268    2270#  2270    2273#  2273
          2275#  2275    2277#  2277    2279#  2279    2281#  2281    2284#  2284    2286#  2286    2288#  2288
          2290#  2290    2292#  2292    2295#  2295    2297#  2297    2299#  2299    2301#  2301    2303#  2303
          2306#  2306    2308#  2308    2310#  2310    2312#  2312    2314#  2314    2317#  2317    2319#  2319
          2321#  2321    2323#  2323    2325#  2325    2328#  2328    2330#  2330    2332#  2332    2334#  2334
          2336#  2336    2339#  2339    2341#  2341    2343#  2343    2345#  2345    2347#  2347    2350#  2350
          2352#  2352    2354#  2354    2356#  2356    2358#  2358    2361#  2361    2363#  2363    2365#  2365
          2367#  2367    2369#  2369    2374#  2374    2379#  2379
$CHR       170#   170     172#   172     174#   174     177#   177     179#   179     181#   181     184#   184
           186#   186     188#   188     191#   191     193#   193     195#   195     288#   288     290#   290
           295#   295     297#   297     299#   299     301#   301     303#   303     305#   305     310#   310
           312#   312     314#   314     316#   316     318#   318     320#   320     325#   325     327#   327
           329#   329     331#   331     333#   333     335#   335     340#   340     342#   342     344#   344
           346#   346     348#   348     350#   350     353#   353     593#   593     595#   595     597#   597
           599#   599     601#   601     603#   603     605#   605     607#   607     609#   609     611#   611
           613#   613     615#   615     620#   620     622#   622     624#   624     626#   626     628#   628
           630#   630     632#   632     634#   634     636#   636     638#   638     640#   640     642#   642
           647#   647     649#   649     651#   651     653#   653     655#   655     657#   657     659#   659
           661#   661     663#   663     665#   665     667#   667     669#   669     674#   674     676#   676
           678#   678     680#   680     682#   682     684#   684     686#   686     688#   688     690#   690
           692#   692     694#   694     696#   696     701#   701     703#   703     705#   705     707#   707
```

```
.LJ
.LJ
.LL
.LM
.LN
.LO
.LO
```

```
           709#   709     711#   711     713#   713     715#   715     717#   717     719#   719     721#   721
```

```
1343

 709#     709      711#     711      713#     713      715#     715      717#     717      719#     719      721#     721     SEQ 1356
 723#     723      728#     728      730#     730      732#     732      734#     734      736#     736      738#     738
 740#     740      742#     742      744#     744      746#     746      748#     748      750#     750      755#     755
 757#     757      759#     759      761#     761      763#     763      765#     765      767#     767      769#     769
 771#     771      773#     773      775#     775      777#     777      782#     782      784#     784      786#     786
 788#     788      790#     790      792#     792      794#     794      796#     796      798#     798      800#     800
 802#     802      804#     804      809#     809      811#     811      813#     813      815#     815      817#     817
 819#     819      821#     821      823#     823      825#     825      827#     827      829#     829      831#     831
 836#     836      838#     838      840#     840      842#     842      844#     844      846#     846      848#     848
 850#     850      852#     852      854#     854      856#     856      858#     858      863#     863      865#     865
 867#     867      869#     869      871#     871      873#     873      875#     875      877#     877      879#     879
 881#     881      883#     883      885#     885      890#     890      892#     892      894#     894      896#     896
 898#     898      900#     900      902#     902      904#     904      906#     906      908#     908      910#     910
 912#     912      917#     917      919#     919      921#     921      923#     923      925#     925      927#     927
 929#     929      931#     931      933#     933      935#     935      937#     937      939#     939      944#     944
 946#     946      948#     948      950#     950      952#     952      954#     954      956#     956      958#     958
 960#     960      962#     962      964#     964      966#     966      971#     971      973#     973      975#     975
 977#     977      979#     979      981#     981      983#     983      985#     985      987#     987      989#     989
 991#     991      993#     993      998#     998     1000#    1000     1002#    1002     1004#    1004     1006#    1006
1008#    1008     1010#    1010     1012#    1012     1014#    1014     1016#    1016     1018#    1018     1020#    1020
1115#    1115     1117#    1117     1119#    1119     1121#    1121     1123#    1123     1125#    1125     1127#    1127
1129#    1129     1131#    1131     1133#    1133     1135#    1135     1137#    1137     1139#    1139     1141#    1141
1143#    1143     1145#    1145     1147#    1147     1149#    1149     1151#    1151     1153#    1153     1155#    1155
1157#    1157     1159#    1159     1161#    1161     1163#    1163     1165#    1165     1167#    1167     1169#    1169
1171#    1171     1173#    1173     1175#    1175     1177#    1177     1179#    1179     1182#    1182     1184#    1184
1186#    1186     1188#    1188     1190#    1190     1192#    1192     1194#    1194     1196#    1196     1198#    1198
1200#    1200     1202#    1202     1204#    1204     1206#    1206     1208#    1208     1210#    1210     1212#    1212
1214#    1214     1216#    1216     1218#    1218     1220#    1220     1222#    1222     1224#    1224     1226#    1226
1228#    1228     1230#    1230     1232#    1232     1234#    1234     1236#    1236     1238#    1238     1240#    1240
1242#    1242     1244#    1244     1247#    1247     1249#    1249     1251#    1251     1253#    1253     1255#    1255
1257#    1257     1259#    1259     1261#    1261     1263#    1263     1265#    1265     1267#    1267     1269#    1269
1271#    1271     1273#    1273     1275#    1275     1277#    1277     1279#    1279     1281#    1281     1283#    1283
1285#    1285     1287#    1287     1289#    1289     1291#    1291     1293#    1293     1295#    1295     1297#    1297
1299#    1299     1301#    1301     1303#    1303     1305#    1305     1307#    1307     1309#    1309     1312#    1312
1314#    1314     1316#    1316     1318#    1318     1320#    1320     1322#    1322     1324#    1324     1326#    1326
1328#    1328     1330#    1330     1332#    1332     1334#    1334     1336#    1336     1338#    1338     1340#    1340
1342#    1342     1344#    1344     1346#    1346     1348#    1348     1350#    1350     1352#    1352     1354#    1354
1356#    1356     1358#    1358     1360#    1360     1362#    1362     1364#    1364     1366#    1366     1368#    1368
1370#    1370     1372#    1372     1374#    1374     1377#    1377     1379#    1379     1473#    1473     1475#    1475
1477#    1477     1479#    1479     1481#    1481     1483#    1483     1485#    1485     1487#    1487     1489#    1489
1491#    1491     1493#    1493     1495#    1495     1497#    1497     1499#    1499     1501#    1501     1503#    1503
1505#    1505     1507#    1507     1509#    1509     1511#    1511     1513#    1513     1515#    1515     1517#    1517
1519#    1519     1521#    1521     1523#    1523     1525#    1525     1527#    1527     1529#    1529     1531#    1531
1533#    1533     1535#    1535     1537#    1537     1539#    1539     1541#    1541     1543#    1543     1545#    1545
1547#    1547     1549#    1549     1551#    1551     1553#    1553     1555#    1555     1557#    1557     1559#    1559
1561#    1561     1563#    1563     1565#    1565     1567#    1567     1569#    1569     1571#    1571     1573#    1573
1576#    1576     1578#    1578     1672#    1672     1674#    1674     1676#    1676     1678#    [illegible]     1680#    1680
1682#    1682     1684#    1684     1686#    1686     1688#    1688     1690#    1690     1692#    [illegible]     1694#    1694
1696#    1696     1698#    1698     1700#    1700     1702#    1702     1704#    1704     1706#    1706     1708#    1708
1710#    1710     1712#    1712     1714#    1714     1716#    1716     1718#    1718     1720#    1720     1722#    1722
1724#    1724     1726#    1726     1728#    1728     1730#    1730     1732#    1732     1734#    1734     1736#    1736
1738#    1738     1740#    1740     1742#    1742     1744#    1744     1746#    1746     1748#    1748     1750#    1750
1752#    1752     1754#    1754     1756#    1756     1758#    1758     1760#    1760     1762#    1762     1764#    1764
```

```
1344    1766#   1766    1768#   1768    1770#   1770    1772#   1772    1775#   1775    1777#   1777    1870#   1870    SEQ 1357
        1872#   1872    1874#   1874    1876#   1876    1878#   1878    1880#   1880    1882#   1882    1884#   1884
        1886#   1886    1888#   1888    1890#   1890    1892#   1892    1894#   1894    1896#   1896    1898#   1898
        1900#   1900    1902#   1902    1904#   1904    1906#   1906    1908#   1908    1910#   1910    1912#   1912
        1914#   1914    1916#   1916    1918#   1918    1920#   1920    1922#   1922    1924#   1924    1926#   1926
        1928#   1928    1930#   1930    1932#   1932    1934#   1934    1936#   1936    1938#   1938    1940#   1940
        1942#   1942    1944#   1944    1946#   1946    1948#   1948    1950#   1950    1952#   1952    1954#   1954
        1956#   1956    1958#   1958    1960#   1960    1962#   1962    1964#   1964    1966#   1966    1968#   1968
        1970#   1970    1973#   1973    2045#   2045    2047#   2047    2050#   2050    2052#   2052    2054#   2054
        2056#   2056    2058#   2058    2061#   2061    2063#   2063    2065#   2065    2067#   2067    2070#   2070
        2072#   2072    2074#   2074    2076#   2076    2079#   2079    2081#   2081    2083#   2083    2085#   2085
        2088#   2088    2090#   2090    2092#   2092    2094#   2094    2097#   2097    2099#   2099    2101#   2101
        2103#   2103    2106#   2106    2108#   2108    2110#   2110    2112#   2112    2115#   2115    2117#   2117
        2119#   2119    2121#   2121    2124#   2124    2126#   2126    2128#   2128    2130#   2130    2133#   2133
        2135#   2135    2137#   2137    2139#   2139    2142#   2142    2144#   2144    2146#   2146    2148#   2148
        2151#   2151    2153#   2153    2155#   2155    2157#   2157    2160#   2160    2162#   2162    2164#   2164
        2166#   2166    2169#   2169    2171#   2171    2173#   2173    2175#   2175    2178#   2178    2180#   2180
        2182#   2182    2184#   2184    2187#   2187    2189#   2189    2191#   2191    2193#   2193    2196#   2196
        2198#   2198    2200#   2200    2202#   2202    2204#   2204    2207#   2207    2209#   2209    2211#   2211
        2213#   2213    2215#   2215    2218#   2218    2220#   2220    2222#   2222    2224#   2224    2226#   2226
        2229#   2229    2231#   2231    2233#   2233    2235#   2235    2237#   2237    2240#   2240    2242#   2242
        2244#   2244    2246#   2246    2248#   2248    2251#   2251    2253#   2253    2255#   2255    2257#   2257
        2259#   2259    2262#   2262    2264#   2264    2266#   2266    2268#   2268    2270#   2270    2273#   2273
        2275#   2275    2277#   2277    2279#   2279    2281#   2281    2284#   2284    2286#   2286    2288#   2288
        2290#   2290    2292#   2292    2295#   2295    2297#   2297    2299#   2299    2301#   2301    2303#   2303
        2306#   2306    2308#   2308    2310#   2310    2312#   2312    2314#   2314    2317#   2317    2319#   2319
        2321#   2321    2323#   2323    2325#   2325    2328#   2328    2330#   2330    2332#   2332    2334#   2334
        2336#   2336    2339#   2339    2341#   2341    2343#   2343    2345#   2345    2347#   2347    2350#   2350
        2352#   2352    2354#   2354    2356#   2356    2358#   2358    2361#   2361    2363#   2363    2365#   2365
        2367#   2367    2369#   2369    2374#   2374    2379#   2379
$GARG    170#    170     172#    172     174#    174     177#    177     179#    179     181#    181     184#    184
         186#    186     188#    188     191#    191     193#    193     195#    195     288#    288     290#    290
         295#    295     297#    297     299#    299     301#    301     303#    303     305#    305     310#    310
         312#    312     314#    314     316#    316     318#    318     320#    320     325#    325     327#    327
         329#    329     331#    331     333#    333     335#    335     340#    340     342#    342     344#    344
         346#    346     348#    348     350#    350     353#    353     593#    593     595#    595     597#    597
         599#    599     601#    601     603#    603     605#    605     607#    607     609#    609     611#    611
         613#    613     615#    615     620#    620     622#    622     624#    624     626#    626     628#    628
         630#    630     632#    632     634#    634     636#    636     638#    638     640#    640     642#    642
         647#    647     649#    649     651#    651     653#    653     655#    655     657#    657     659#    659
         661#    661     663#    663     665#    665     667#    667     669#    669     674#    674     676#    676
         678#    678     680#    680     682#    682     684#    684     686#    686     688#    688     690#    690
         692#    692     694#    694     696#    696     701#    701     703#    703     705#    705     707#    707
         709#    709     711#    711     713#    713     715#    715     717#    717     719#    719     721#    721
         723#    723     728#    728     730#    730     732#    732     734#    734     736#    736     738#    738
         740#    740     742#    742     744#    744     746#    746     748#    748     750#    750     755#    755
         757#    757     759#    759     761#    761     763#    763     765#    765     767#    767     769#    769
         771#    771     773#    773     775#    775     777#    777     782#    782     784#    784     786#    786
         788#    788     790#    790     792#    792     794#    794     796#    796     798#    798     800#    800
         802#    802     804#    804     809#    809     811#    811     813#    813     815#    815     817#    817
         819#    819     821#    821     823#    823     825#    825     827#    827     829#    829     831#    831
         836#    836     838#    838     840#    840     842#    842     844#    844     846#    846     848#    848
         850#    850     852#    852     854#    854     856#    856     858#    858     863#    863     865#    865
```

```
 867#    867     869#    869     871#    871     873#    873     875#    875     877#    877     879#    879
 881#    881     883#    883     885#    885     890#    890     892#    892     894#    894     896#    896
 898#    898     900#    900     902#    902     904#    904     906#    906     908#    908     910#    910
 912#    912     917#    917     919#    919     921#    921     923#    923     925#    925     927#    927
 929#    929     931#    931     933#    933     935#    935     937#    937     939#    939     944#    944
 946#    946     948#    948     950#    950     952#    952     954#    954     956#    956     958#    958
 960#    960     962#    962     964#    964     966#    966     971#    971     973#    973     975#    975
 977#    977     979#    979     981#    981     983#    983     985#    985     987#    987     989#    989
 991#    991     993#    993     998#    998    1000#   1000    1002#   1002    1004#   1004    1006#   1006
1008#   1008    1010#   1010    1012#   1012    1014#   1014    1016#   1016    1018#   1018    1020#   1020
1115#   1115    1117#   1117    1119#   1119    1121#   1121    1123#   1123    1125#   1125    1127#   1127
1129#   1129    1131#   1131    1133#   1133    1135#   1135    1137#   1137    1139#   1139    1141#   1141
1143#   1143    1145#   1145    1147#   1147    1149#   1149    1151#   1151    1153#   1153    1155#   1155
1157#   1157    1159#   1159    1161#   1161    1163#   1163    1165#   1165    1167#   1167    1169#   1169
1171#   1171    1173#   1173    1175#   1175    1177#   1177    1179#   1179    1182#   1182    1184#   1184
1186#   1186    1188#   1188    1190#   1190    1192#   1192    1194#   1194    1196#   1196    1198#   1198
1200#   1200    1202#   1202    1204#   1204    1206#   1206    1208#   1208    1210#   1210    1212#   1212
1214#   1214    1216#   1216    1218#   1218    1220#   1220    1222#   1222    1224#   1224    1226#   1226
1228#   1228    1230#   1230    1232#   1232    1234#   1234    1236#   1236    1238#   1238    1240#   1240
1242#   1242    1244#   1244    1247#   1247    1249#   1249    1251#   1251    1253#   1253    1255#   1255
1257#   1257    1259#   1259    1261#   1261    1263#   1263    1265#   1265    1267#   1267    1269#   1269
1271#   1271    1273#   1273    1275#   1275    1277#   1277    1279#   1279    1281#   1281    1283#   1283
1285#   1285    1287#   1287    1289#   1289    1291#   1291    1293#   1293    1295#   1295    1297#   1297
1299#   1299    1301#   1301    1303#   1303    1305#   1305    1307#   1307    1309#   1309    1312#   1312
1314#   1314    1316#   1316    1318#   1318    1320#   1320    1322#   1322    1324#   1324    1326#   1326
1328#   1328    1330#   1330    1332#   1332    1334#   1334    1336#   1336    1338#   1338    1340#   1340
1342#   1342    1344#   1344    1346#   1346    1348#   1348    1350#   1350    1352#   1352    1354#   1354
1356#   1356    1358#   1358    1360#   1360    1362#   1362    1364#   1364    1366#   1366    1368#   1368
1370#   1370    1372#   1372    1374#   1374    1377#   1377    1379#   1379    1473#   1473    1475#   1475
1477#   1477    1479#   1479    1481#   1481    1483#   1483    1485#   1485    1487#   1487    1489#   1489
1491#   1491    1493#   1493    1495#   1495    1497#   1497    1499#   1499    1501#   1501    1503#   1503
1505#   1505    1507#   1507    1509#   1509    1511#   1511    1513#   1513    1515#   1515    1517#   1517
1519#   1519    1521#   1521    1523#   1523    1525#   1525    1527#   1527    1529#   1529    1531#   1531
1533#   1533    1535#   1535    1537#   1537    1539#   1539    1541#   1541    1543#   1543    1545#   1545
1547#   1547    1549#   1549    1551#   1551    1553#   1553    1555#   1555    1557#   1557    1559#   1559
1561#   1561    1563#   1563    1565#   1565    1567#   1567    1569#   1569    1571#   1571    1573#   1573
1576#   1576    1578#   1578    1672#   1672    1674#   1674    1676#   1676    1678#   1678    1680#   1680
1682#   1682    1684#   1684    1686#   1686    1688#   1688    1690#   1690    1692#   1692    1694#   1694
1696#   1696    1698#   1698    1700#   1700    1702#   1702    1704#   1704    1706#   1706    1708#   1708
1710#   1710    1712#   1712    1714#   1714    1716#   1716    1718#   1718    1720#   1720    1722#   1722
1724#   1724    1726#   1726    1728#   1728    1730#   1730    1732#   1732    1734#   1734    1736#   1736
1738#   1738    1740#   1740    1742#   1742    1744#   1744    1746#   1746    1748#   1748    1750#   1750
1752#   1752    1754#   1754    1756#   1756    1758#   1758    1760#   1760    1762#   1762    1764#   1764
1766#   1766    1768#   1768    1770#   1770    1772#   1772    1775#   1775    1777#   1777    1870#   1870
1872#   1872    1874#   1874    1876#   1876    1878#   1878    1880#   1880    1882#   1882    1884#   1884
1886#   1886    1888#   1888    1890#   1890    1892#   1892    1894#   1894    1896#   1896    1898#   1898
1900#   1900    1902#   1902    1904#   1904    1906#   1906    1908#   1908    1910#   1910    1912#   1912
1914#   1914    1916#   1916    1918#   1918    1920#   1920    1922#   1922    1924#   1924    1926#   1926
1928#   1928    1930#   1930    1932#   1932    1934#   1934    1936#   1936    1938#   1938    1940#   1940
1942#   1942    1944#   1944    1946#   1946    1948#   1948    1950#   1950    1952#   1952    1954#   1954
1956#   1956    1958#   1958    1960#   1960    1962#   1962    1964#   1964    1966#   1966    1968#   1968
1970#   1970    1973#   1973    2045#   2045    2047#   2047    2050#   2050    2052#   2052    2054#   2054
2056#   2056    2058#   2058    2061#   2061    2063#   2063    2065#   2065    2067#   2067    2070#   2070
```

```
         2072#   2072    2074#   2074    2076#   2076    2079#   2079    2081#   2081    2083#   2083    2085#   2085
         2088#   2088    2090#   2090    2092#   2092    2094#   2094    2097#   2097    2099#   2099    2101#   2101
         2103#   2103    2106#   2106    2108#   2108    2110#   2110    2112#   2112    2115#   2115    2117#   2117
         2119#   2119    2121#   2121    2124#   2124    2126#   2126    2128#   2128    2130#   2130    2133#   2133
         2135#   2135    2137#   2137    2139#   2139    2142#   2142    2144#   2144    2146#   2146    2148#   2148
         2151#   2151    2153#   2153    2155#   2155    2157#   2157    2160#   2160    2162#   2162    2164#   2164
         2166#   2166    2169#   2169    2171#   2171    2173#   2173    2175#   2175    2178#   2178    2180#   2180
         2182#   2182    2184#   2184    2187#   2187    2189#   2189    2191#   2191    2193#   2193    2196#   2196
         2198#   2198    2200#   2200    2202#   2202    2204#   2204    2207#   2207    2209#   2209    2211#   2211
         2213#   2213    2215#   2215    2218#   2218    2220#   2220    2222#   2222    2224#   2224    2226#   2226
         2229#   2229    2231#   2231    2233#   2233    2235#   2235    2237#   2237    2240#   2240    2242#   2242
         2244#   2244    2246#   2246    2248#   2248    2251#   2251    2253#   2253    2255#   2255    2257#   2257
         2259#   2259    2262#   2262    2264#   2264    2266#   2266    2268#   2268    2270#   2270    2273#   2273
         2275#   2275    2277#   2277    2279#   2279    2281#   2281    2284#   2284    2286#   2286    2288#   2288
         2290#   2290    2292#   2292    2295#   2295    2297#   2297    2299#   2299    2301#   2301    2303#   2303
         2306#   2306    2308#   2308    2310#   2310    2312#   2312    2314#   2314    2317#   2317    2319#   2319
         2321#   2321    2323#   2323    2325#   2325    2328#   2328    2330#   2330    2332#   2332    2334#   2334
         2336#   2336    2339#   2339    2341#   2341    2343#   2343    2345#   2345    2347#   2347    2350#   2350
         2352#   2352    2354#   2354    2356#   2356    2358#   2358    2361#   2361    2363#   2363    2365#   2365
         2367#   2367    2369#   2369    2374#   2374    2379#   2379
%ADDR     170#    170     172     172#    174     174#    177     177#    179     179#    181     181#    184     184#
          186     186#    188     188#    191     191#    193     193#    195     195#    288#    288     290     290#
          295     295#    297     297#    299     299#    301     301#    303     303#    305     305#    310     310#
          312     312#    314     314#    316     316#    318     318#    320     320#    325     325#    327     327#
          329     329#    331     331#    333     333#    335     335#    340     340#    342     342#    344     344#
          346     346#    348     348#    350     350#    353     353#    593#    593     595     595#    597     597#
          599     599#    601     601#    603     603#    605     605#    607     607#    609     609#    611     611#
          613     613#    615     615#    620#    620     622     622#    624     624#    626     626#    628     628#
          630     630#    632     632#    634     634#    636     636#    638     638#    640     640#    642     642#
          647#    647     649     649#    651     651#    653     653#    655     655#    657     657#    659     659#
          661     661#    663     663#    665     665#    667     667#    669     669#    674#    674     676     676#
          678     678#    680     680#    682     682#    684     684#    686     686#    688     688#    690     690#
          692     692#    694     694#    696     696#    701#    701     703     703#    705     705#    707     707#
          709     709#    711     711#    713     713#    715     715#    717     717#    719     719#    721     721#
          723     723#    728#    728     730     730#    732     732#    734     734#    736     736#    738     738#
          740     740#    742     742#    744     744#    746     746#    748     748#    750     750#    755#    755
          757     757#    759     759#    761     761#    763     763#    765     765#    767     767#    769     769#
          771     771#    773     773#    775     775#    777     777#    782#    782     784     784#    786     786#
          788     788#    790     790#    792     792#    794     794#    796     796#    798     798#    800     800#
          802     802#    804     804#    809#    809     811     811#    813     813#    815     815#    817     817#
          819     819#    821     821#    823     823#    825     825#    827     827#    829     829#    831     831#
          836#    836     838     838#    840     840#    842     842#    844     844#    846     846#    848     848#
          850     850#    852     852#    854     854#    856     856#    858     858#    863#    863     865     865#
          867     867#    869     869#    871     871#    873     873#    875     875#    877     877#    879     879#
          881     881#    883     883#    885     885#    890#    890     892     892#    894     894#    896     896#
          898     898#    900     900#    902     902#    904     904#    906     906#    908     908#    910     910#
          912     912#    917#    917     919     919#    921     921#    923     923#    925     925#    927     927#
          929     929#    931     931#    933     933#    935     935#    937     937#    939     939#    944#    944
          946     946#    948     948#    950     950#    952     952#    954     954#    956     956#    958     958#
          960     960#    962     962#    964     964#    966     966#    971#    971     973     973#    975     975#
          977     977#    979     979#    981     981#    983     983#    985     985#    987     987#    989     989#
          991     991#    993     993#    998#    998    1000    1000#   1002    1002#   1004    1004#   1006    1006#
         1008    1008#   1010    1010#   1012    1012#   1014    1014#   1016    1016#   1018    1018#   1020    1020#
```

```
Q 1347   1115#  1115   1117   1117#  1119   1119#  1121   1121#  1123   1123#  1125   1125#  1127   1127#   SEQ 1360
         1129   1129#  1131   1131#  1133   1133#  1135   1135#  1137   1137#  1139   1139#  1141   1141#
         1143   1143#  1145   1145#  1147   1147#  1149   1149#  1151   1151#  1153   1153#  1155   1155#
         1157   1157#  1159   1159#  1161   1161#  1163   1163#  1165   1165#  1167   1167#  1169   1169#
         1171   1171#  1173   1173#  1175   1175#  1177   1177#  1179   1179#  1182   1182#  1184   1184#
         1186   1186#  1188   1188#  1190   1190#  1192   1192#  1194   1194#  1196   1196#  1198   1198#
         1200   1200#  1202   1202#  1204   1204#  1206   1206#  1208   1208#  1210   1210#  1212   1212#
         1214   1214#  1216   1216#  1218   1218#  1220   1220#  1222   1222#  1224   1224#  1226   1226#
         1228   1228#  1230   1230#  1232   1232#  1234   1234#  1236   1236#  1238   1238#  1240   1240#
         1242   1242#  1244   1244#  1247   1247#  1249   1249#  1251   1251#  1253   1253#  1255   1255#
         1257   1257#  1259   1259#  1261   1261#  1263   1263#  1265   1265#  1267   1267#  1269   1269#
         1271   1271#  1273   1273#  1275   1275#  1277   1277#  1279   1279#  1281   1281#  1283   1283#
         1285   1285#  1287   1287#  1289   1289#  1291   1291#  1293   1293#  1295   1295#  1297   1297#
         1299   1299#  1301   1301#  1303   1303#  1305   1305#  1307   1307#  1309   1309#  1312   1312#
         1314   1314#  1316   1316#  1318   1318#  1320   1320#  1322   1322#  1324   1324#  1326   1326#
         1328   1328#  1330   1330#  1332   1332#  1334   1334#  1336   1336#  1338   1338#  1340   1340#
         1342   1342#  1344   1344#  1346   1346#  1348   1348#  1350   1350#  1352   1352#  1354   1354#
         1356   1356#  1358   1358#  1360   1360#  1362   1362#  1364   1364#  1366   1366#  1368   1368#
         1370   1370#  1372   1372#  1374   1374#  1377   1377#  1379   1379#  1473#  1473   1475   1475#
         1477   1477#  1479   1479#  1481   1481#  1483   1483#  1485   1485#  1487   1487#  1489   1489#
         1491   1491#  1493   1493#  1495   1495#  1497   1497#  1499   1499#  1501   1501#  1503   1503#
         1505   1505#  1507   1507#  1509   1509#  1511   1511#  1513   1513#  1515   1515#  1517   1517#
         1519   1519#  1521   1521#  1523   1523#  1525   1525#  1527   1527#  1529   1529#  1531   1531#
         1533   1533#  1535   1535#  1537   1537#  1539   1539#  1541   1541#  1543   1543#  1545   1545#
         1547   1547#  1549   1549#  1551   1551#  1553   1553#  1555   1555#  1557   1557#  1559   1559#
         1561   1561#  1563   1563#  1565   1565#  1567   1567#  1569   1569#  1571   1571#  1573   1573#
         1576   1576#  1578   1578#  1672#  1672   1674   1674#  1676   1676#  1678   1678#  1680   1680#
         1682   1682#  1684   1684#  1686   1686#  1688   1688#  1690   1690#  1692   1692#  1694   1694#
         1696   1696#  1698   1698#  1700   1700#  1702   1702#  1704   1704#  1706   1706#  1708   1708#
         1710   1710#  1712   1712#  1714   1714#  1716   1716#  1718   1718#  1720   1720#  1722   1722#
         1724   1724#  1726   1726#  1728   1728#  1730   1730#  1732   1732#  1734   1734#  1736   1736#
         1738   1738#  1740   1740#  1742   1742#  1744   1744#  1746   1746#  1748   1748#  1750   1750#
         1752   1752#  1754   1754#  1756   1756#  1758   1758#  1760   1760#  1762   1762#  1764   1764#
         1766   1766#  1768   1768#  1770   1770#  1772   1772#  1775   1775#  1777   1777#  1870#  1870
         1872   1872#  1874   1874#  1876   1876#  1878   1878#  1880   1880#  1882   1882#  1884   1884#
         1886   1886#  1888   1888#  1890   1890#  1892   1892#  1894   1894#  1896   1896#  1898   1898#
         1900   1900#  1902   1902#  1904   1904#  1906   1906#  1908   1908#  1910   1910#  1912   1912#
         1914   1914#  1916   1916#  1918   1918#  1920   1920#  1922   1922#  1924   1924#  1926   1926#
         1928   1928#  1930   1930#  1932   1932#  1934   1934#  1936   1936#  1938   1938#  1940   1940#
         1942   1942#  1944   1944#  1946   1946#  1948   1948#  1950   1950#  1952   1952#  1954   1954#
         1956   1956#  1958   1958#  1960   1960#  1962   1962#  1964   1964#  1966   1966#  1968   1968#
         1970   1970#  1973   1973#  2045#  2045   2047   2047#  2050#  2050   2052   2052#  2054   2054#
         2056   2056#  2058   2058#  2061   2061#  2063   2063#  2065   2065#  2067   2067#  2070   2070#
         2072   2072#  2074   2074#  2076   2076#  2079   2079#  2081   2081#  2083   2083#  2085   2085#
         2088   2088#  2090   2090#  2092   2092#  2094   2094#  2097   2097#  2099   2099#  2101   2101#
         2103   2103#  2106   2106#  2108   2108#  2110   2110#  2112   2112#  2115   2115#  2117   2117#
         2119   2119#  2121   2121#  2124   2124#  2126   2126#  2128   2128#  2130   2130#  2133   2133#
         2135   2135#  2137   2137#  2139   2139#  2142   2142#  2144   2144#  2146   2146#  2148   2148#
         2151   2151#  2153   2153#  2155   2155#  2157   2157#  2160   2160#  2162   2162#  2164   2164#
         2166   2166#  2169   2169#  2171   2171#  2173   2173#  2175   2175#  2178   2178#  2180   2180#
         2182   2182#  2184   2184#  2187   2187#  2189   2189#  2191   2191#  2193   2193#  2196#  2196
         2198   2198#  2200   2200#  2202   2202#  2204   2204#  2207   2207#  2209   2209#  2211   2211#
         2213   2213#  2215   2215#  2218   2218#  2220   2220#  2222   2222#  2224   2224#  2226   2226#
```

```
EQ 1348 2229   2229#  2231   2231#  2233   2233#  2235   2235#  2237   2237#  2240   2240#  2242   2242#   SEQ 1361
        2244   2244#  2246   2246#  2248   2248#  2251   2251#  2253   2253#  2255   2255#  2257   2257#
        2259   2259#  2262   2262#  2264   2264#  2266   2266#  2268   2268#  2270   2270#  2273   2273#
        2275   2275#  2277   2277#  2279   2279#  2281   2281#  2284   2284#  2286   2286#  2288   2288#
        2290   2290#  2292   2292#  2295   2295#  2297   2297#  2299   2299#  2301   2301#  2303   2303#
        2306   2306#  2308   2308#  2310   2310#  2312   2312#  2314   2314#  2317   2317#  2319   2319#
        2321   2321#  2323   2323#  2325   2325#  2328   2328#  2330   2330#  2332   2332#  2334   2334#
        2336   2336#  2339   2339#  2341   2341#  2343   2343#  2345   2345#  2347   2347#  2350   2350#
        2352   2352#  2354   2354#  2356   2356#  2358   2358#  2361   2361#  2363   2363#  2365   2365#
        2367   2367#  2369   2369#  2374#  2374   2379#  2379
%ML      170#   170    172#   172    174#   174    177#   177    179#   179    181#   181    184#   184
         186#   186    188#   188    191#   191    193#   193    195#   195    288#   288    290#   290
         295#   295    297#   297    299#   299    301#   301    303#   303    305#   305    310#   310
         312#   312    314#   314    316#   316    318#   318    320#   320    325#   325    327#   327
         329#   329    331#   331    333#   333    335#   335    340#   340    342#   342    344#   344
         346#   346    348#   348    350#   350    353#   353    593#   593    595#   595    597#   597
         599#   599    601#   601    603#   603    605#   605    607#   607    609#   609    611#   611
         613#   613    615#   615    620#   620    622#   622    624#   624    626#   626    628#   628
         630#   630    632#   632    634#   634    636#   636    638#   638    640#   640    642#   642
         647#   647    649#   649    651#   651    653#   653    655#   655    657#   657    659#   659
         661#   661    663#   663    665#   665    667#   667    669#   669    674#   674    676#   676
         678#   678    680#   680    682#   682    684#   684    686#   686    688#   688    690#   690
         692#   692    694#   694    696#   696    701#   701    703#   703    705#   705    707#   707
         709#   709    711#   711    713#   713    715#   715    717#   717    719#   719    721#   721
         723#   723    728#   728    730#   730    732#   732    734#   734    736#   736    738#   738
         740#   740    742#   742    744#   744    746#   746    748#   748    750#   750    755#   755
         757#   757    759#   759    761#   761    763#   763    765#   765    767#   767    769#   769
         771#   771    773#   773    775#   775    777#   777    782#   782    784#   784    786#   786
         788#   788    790#   790    792#   792    794#   794    796#   796    798#   798    800#   800
         802#   802    804#   804    809#   809    811#   811    813#   813    815#   815    817#   817
         819#   819    821#   821    823#   823    825#   825    827#   827    829#   829    831#   831
         836#   836    838#   838    840#   840    842#   842    844#   844    846#   846    848#   848
         850#   850    852#   852    854#   854    856#   856    858#   858    863#   863    865#   865
         867#   867    869#   869    871#   871    873#   873    875#   875    877#   877    879#   879
         881#   881    883#   883    885#   885    890#   890    892#   892    894#   894    896#   896
         898#   898    900#   900    902#   902    904#   904    906#   906    908#   908    910#   910
         912#   912    917#   917    919#   919    921#   921    923#   923    925#   925    927#   927
         929#   929    931#   931    933#   933    935#   935    937#   937    939#   939    944#   944
         946#   946    948#   948    950#   950    952#   952    954#   954    956#   956    958#   958
         960#   960    962#   962    964#   964    966#   966    971#   971    973#   973    975#   975
         977#   977    979#   979    981#   981    983#   983    985#   985    987#   987    989#   989
         991#   991    993#   993    998#   998   1000#  1000   1002#  1002   1004#  1004   1006#  1006
        1008#  1008   1010#  1010   1012#  1012   1014#  1014   1016#  1016   1018#  1018   1020#  1020
        1115#  1115   1117#  1117   1119#  1119   1121#  1121   1123#  1123   1125#  1125   1127#  1127
        1129#  1129   1131#  1131   1133#  1133   1135#  1135   1137#  1137   1139#  1139   1141#  1141
        1143#  1143   1145#  1145   1147#  1147   1149#  1149   1151#  1151   1153#  1153   1155#  1155
        1157#  1157   1159#  1159   1161#  1161   1163#  1163   1165#  1165   1167#  1167   1169#  1169
        1171#  1171   1173#  1173   1175#  1175   1177#  1177   1179#  1179   1182#  1182   1184#  1184
        1186#  1186   1188#  1188   1190#  1190   1192#  1192   1194#  1194   1196#  1196   1198#  1198
        1200#  1200   1202#  1202   1204#  1204   1206#  1206   1208#  1208   1210#  1210   1212#  1212
        1214#  1214   1216#  1216   1218#  1218   1220#  1220   1222#  1222   1224#  1224   1226#  1226
        1228#  1228   1230#  1230   1232#  1232   1234#  1234   1236#  1236   1238#  1238   1240#  1240
        1242#  1242   1244#  1244   1247#  1247   1249#  1249   1251#  1251   1253#  1253   1255#  1255
```

```
EQ 1349   1257#  1257   1259#  1259   1261#  1261   1263#  1263   1265#  1265   1267#  1267   1269#  1269
          1271#  1271   1273#  1273   1275#  1275   1277#  1277   1279#  1279   1281#  1281   1283#  1283     SEQ 1362
          1285#  1285   1287#  1287   1289#  1289   1291#  1291   1293#  1293   1295#  1295   1297#  1297
          1299#  1299   1301#  1301   1303#  1303   1305#  1305   1307#  1307   1309#  1309   1312#  1312
          1314#  1314   1316#  1316   1318#  1318   1320#  1320   1322#  1322   1324#  1324   1326#  1326
          1328#  1328   1330#  1330   1332#  1332   1334#  1334   1336#  1336   1338#  1338   1340#  1340
          1342#  1342   1344#  1344   1346#  1346   1348#  1348   1350#  1350   1352#  1352   1354#  1354
          1356#  1356   1358#  1358   1360#  1360   1362#  1362   1364#  1364   1366#  1366   1368#  1368
          1370#  1370   1372#  1372   1374#  1374   1377#  1377   1379#  1379   1473#  1473   1475#  1475
          1477#  1477   1479#  1479   1481#  1481   1483#  1483   1485#  1485   1487#  1487   1489#  1489
          1491#  1491   1493#  1493   1495#  1495   1497#  1497   1499#  1499   1501#  1501   1503#  1503
          1505#  1505   1507#  1507   1509#  1509   1511#  1511   1513#  1513   1515#  1515   1517#  1517
          1519#  1519   1521#  1521   1523#  1523   1525#  1525   1527#  1527   1529#  1529   1531#  1531
          1533#  1533   1535#  1535   1537#  1537   1539#  1539   1541#  1541   1543#  1543   1545#  1545
          1547#  1547   1549#  1549   1551#  1551   1553#  1553   1555#  1555   1557#  1557   1559#  1559
          1561#  1561   1563#  1563   1565#  1565   1567#  1567   1569#  1569   1571#  1571   1573#  1573
          1576#  1576   1578#  1578   1672#  1672   1674#  1674   1676#  1676   1678#  1678   1680#  1680
          1682#  1682   1684#  1684   1686#  1686   1688#  1688   1690#  1690   1692#  1692   1694#  1694
          1696#  1696   1698#  1698   1700#  1700   1702#  1702   1704#  1704   1706#  1706   1708#  1708
          1710#  1710   1712#  1712   1714#  1714   1716#  1716   1718#  1718   1720#  1720   1722#  1722
          1724#  1724   1726#  1726   1728#  1728   1730#  1730   1732#  1732   1734#  1734   1736#  1736
          1738#  1738   1740#  1740   1742#  1742   1744#  1744   1746#  1746   1748#  1748   1750#  1750
          1752#  1752   1754#  1754   1756#  1756   1758#  1758   1760#  1760   1762#  1762   1764#  1764
          1766#  1766   1768#  1768   1770#  1770   1772#  1772   1775#  1775   1777#  1777   1870#  1870
          1872#  1872   1874#  1874   1876#  1876   1878#  1878   1880#  1880   1882#  1882   1884#  1884
          1886#  1886   1888#  1888   1890#  1890   1892#  1892   1894#  1894   1896#  1896   1898#  1898
          1900#  1900   1902#  1902   1904#  1904   1906#  1906   1908#  1908   1910#  1910   1912#  1912
          1914#  1914   1916#  1916   1918#  1918   1920#  1920   1922#  1922   1924#  1924   1926#  1926
          1928#  1928   1930#  1930   1932#  1932   1934#  1934   1936#  1936   1938#  1938   1940#  1940
          1942#  1942   1944#  1944   1946#  1946   1948#  1948   1950#  1950   1952#  1952   1954#  1954
          1956#  1956   1958#  1958   1960#  1960   1962#  1962   1964#  1964   1966#  1966   1968#  1968
          1970#  1970   1973#  1973   2045#  2045   2047#  2047   2050#  2050   2052#  2052   2054#  2054
          2056#  2056   2058#  2058   2061#  2061   2063#  2063   2065#  2065   2067#  2067   2070#  2070
          2072#  2072   2074#  2074   2076#  2076   2079#  2079   2081#  2081   2083#  2083   2085#  2085
          2088#  2088   2090#  2090   2092#  2092   2094#  2094   2097#  2097   2099#  2099   2101#  2101
          2103#  2103   2106#  2106   2108#  2108   2110#  2110   2112#  2112   2115#  2115   2117#  2117
          2119#  2119   2121#  2121   2124#  2124   2126#  2126   2128#  2128   2130#  2130   2133#  2133
          2135#  2135   2137#  2137   2139#  2139   2142#  2142   2144#  2144   2146#  2146   2148#  2148
          2151#  2151   2153#  2153   2155#  2155   2157#  2157   2160#  2160   2162#  2162   2164#  2164
          2166#  2166   2169#  2169   2171#  2171   2173#  2173   2175#  2175   2178#  2178   2180#  2180
          2182#  2182   2184#  2184   2187#  2187   2189#  2189   2191#  2191   2193#  2193   2196#  2196
          2198#  2198   2200#  2200   2202#  2202   2204#  2204   2207#  2207   2209#  2209   2211#  2211
          2213#  2213   2215#  2215   2218#  2218   2220#  2220   2222#  2222   2224#  2224   2226#  2226
          2229#  2229   2231#  2231   2233#  2233   2235#  2235   2237#  2237   2240#  2240   2242#  2242
          2244#  2244   2246#  2246   2248#  2248   2251#  2251   2253#  2253   2255#  2255   2257#  2257
          2259#  2259   2262#  2262   2264#  2264   2266#  2266   2268#  2268   2270#  2270   2273#  2273
          2275#  2275   2277#  2277   2279#  2279   2281#  2281   2284#  2284   2286#  2286   2288#  2288
          2290#  2290   2292#  2292   2295#  2295   2297#  2297   2299#  2299   2301#  2301   2303#  2303
          2306#  2306   2308#  2308   2310#  2310   2312#  2312   2314#  2314   2317#  2317   2319#  2319
          2321#  2321   2323#  2323   2325#  2325   2328#  2328   2330#  2330   2332#  2332   2334#  2334
          2336#  2336   2339#  2339   2341#  2341   2343#  2343   2345#  2345   2347#  2347   2350#  2350
          2352#  2352   2354#  2354   2356#  2356   2358#  2358   2361#  2361   2363#  2363   2365#  2365
          2367#  2367   2369#  2369   2374#  2374   2379#  2379
```

```
SEQ 1350  %MR   170#  170    172#  172    174#  174    177#  177    179#  179    181#  181    184#  184   SEQ 1363
                186#  186    188#  188    191#  191    193#  193    195#  195    288#  288    290#  290
                295#  295    297#  297    299#  299    301#  301    303#  303    305#  305    310#  310
                312#  312    314#  314    316#  316    318#  318    320#  320    325#  325    327#  327
                329#  329    331#  331    333#  333    335#  335    340#  340    342#  342    344#  344
                346#  346    348#  348    350#  350    353#  353    593#  593    595#  595    597#  597
                599#  599    601#  601    603#  603    605#  605    607#  607    609#  609    611#  611
                613#  613    615#  615    620#  620    622#  622    624#  624    626#  626    628#  628
                630#  630    632#  632    634#  634    636#  636    638#  638    640#  640    642#  642
                647#  647    649#  649    651#  651    653#  653    655#  655    657#  657    659#  659
                661#  661    663#  663    665#  665    667#  667    669#  669    674#  674    676#  676
                678#  678    680#  680    682#  682    684#  684    686#  686    688#  688    690#  690
                692#  692    694#  694    696#  696    701#  701    703#  703    705#  705    707#  707
                709#  709    711#  711    713#  713    715#  715    717#  717    719#  719    721#  721
                723#  723    728#  728    730#  730    732#  732    734#  734    736#  736    738#  738
                740#  740    742#  742    744#  744    746#  746    748#  748    750#  750    755#  755
                757#  757    759#  759    761#  761    763#  763    765#  765    767#  767    769#  769
                771#  771    773#  773    775#  775    777#  777    782#  782    784#  784    786#  786
                788#  788    790#  790    792#  792    794#  794    796#  796    798#  798    800#  800
                802#  802    804#  804    809#  809    811#  811    813#  813    815#  815    817#  817
                819#  819    821#  821    823#  823    825#  825    827#  827    829#  829    831#  831
                836#  836    838#  838    840#  840    842#  842    844#  844    846#  846    848#  848
                850#  850    852#  852    854#  854    856#  856    858#  858    863#  863    865#  865
                867#  867    869#  869    871#  871    873#  873    875#  875    877#  877    879#  879
                881#  881    883#  883    885#  885    890#  890    892#  892    894#  894    896#  896
                898#  898    900#  900    902#  902    904#  904    906#  906    908#  908    910#  910
                912#  912    917#  917    919#  919    921#  921    923#  923    925#  925    927#  927
                929#  929    931#  931    933#  933    935#  935    937#  937    939#  939    944#  944
                946#  946    948#  948    950#  950    952#  952    954#  954    956#  956    958#  958
                960#  960    962#  962    964#  964    966#  966    971#  971    973#  973    975#  975
                977#  977    979#  979    981#  981    983#  983    985#  985    987#  987    989#  989
                991#  991    993#  993    998#  998   1000# 1000   1002# 1002   1004# 1004   1006# 1006
               1008# 1008   1010# 1010   1012# 1012   1014# 1014   1016# 1016   1018# 1018   1020# 1020
               1115# 1115   1117# 1117   1119# 1119   1121# 1121   1123# 1123   1125# 1125   1127# 1127
               1129# 1129   1131# 1131   1133# 1133   1135# 1135   1137# 1137   1139# 1139   1141# 1141
               1143# 1143   1145# 1145   1147# 1147   1149# 1149   1151# 1151   1153# 1153   1155# 1155
               1157# 1157   1159# 1159   1161# 1161   1163# 1163   1165# 1165   1167# 1167   1169# 1169
               1171# 1171   1173# 1173   1175# 1175   1177# 1177   1179# 1179   1182# 1182   1184# 1184
               1186# 1186   1188# 1188   1190# 1190   1192# 1192   1194# 1194   1196# 1196   1198# 1198
               1200# 1200   1202# 1202   1204# 1204   1206# 1206   1208# 1208   1210# 1210   1212# 1212
               1214# 1214   1216# 1216   1218# 1218   1220# 1220   1222# 1222   1224# 1224   1226# 1226
               1228# 1228   1230# 1230   1232# 1232   1234# 1234   1236# 1236   1238# 1238   1240# 1240
               1242# 1242   1244# 1244   1247# 1247   1249# 1249   1251# 1251   1253# 1253   1255# 1255
               1257# 1257   1259# 1259   1261# 1261   1263# 1263   1265# 1265   1267# 1267   1269# 1269
               1271# 1271   1273# 1273   1275# 1275   1277# 1277   1279# 1279   1281# 1281   1283# 1283
               1285# 1285   1287# 1287   1289# 1289   1291# 1291   1293# 1293   1295# 1295   1297# 1297
               1299# 1299   1301# 1301   1303# 1303   1305# 1305   1307# 1307   1309# 1309   1312# 1312
               1314# 1314   1316# 1316   1318# 1318   1320# 1320   1322# 1322   1324# 1324   1326# 1326
               1328# 1328   1330# 1330   1332# 1332   1334# 1334   1336# 1336   1338# 1338   1340# 1340
               1342# 1342   1344# 1344   1346# 1346   1348# 1348   1350# 1350   1352# 1352   1354# 1354
               1356# 1356   1358# 1358   1360# 1360   1362# 1362   1364# 1364   1366# 1365   1368# 1368
               1370# 1370   1372# 1372   1374# 1374   1377# 1377   1379# 1379   1473# 1473   1475# 1475
               1477# 1477   1479# 1479   1481# 1481   1483# 1483   1485# 1485   1487# 1487   1489# 1489
```

```
SEQ 1351  1491#  1491   1493#  1493   1495#  1495   1497#  1497   1499#  1499   1501#  1501   1503#  1503   SEQ 1364
          1505#  1505   1507#  1507   1509#  1509   1511#  1511   1513#  1513   1515#  1515   1517#  1517
          1519#  1519   1521#  1521   1523#  1523   1525#  1525   1527#  1527   1529#  1529   1531#  1531
          1533#  1533   1535#  1535   1537#  1537   1539#  1539   1541#  1541   1543#  1543   1545#  1545
          1547#  1547   1549#  1549   1551#  1551   1553#  1553   1555#  1555   1557#  1557   1559#  1559
          1561#  1561   1563#  1563   1565#  1565   1567#  1567   1569#  1569   1571#  1571   1573#  1573
          1576#  1576   1578#  1578   1672#  1672   1674#  1674   1676#  1676   1678#  1678   1680#  1680
          1682#  1682   1684#  1684   1686#  1686   1688#  1688   1690#  1690   1692#  1692   1694#  1694
          1696#  1696   1698#  1698   1700#  1700   1702#  1702   1704#  1704   1706#  1706   1708#  1708
          1710#  1710   1712#  1712   1714#  1714   1716#  1716   1718#  1718   1720#  1720   1722#  1722
          1724#  1724   1726#  1726   1728#  1728   1730#  1730   1732#  1732   1734#  1734   1736#  1736
          1738#  1738   1740#  1740   1742#  1742   1744#  1744   1746#  1746   1748#  1748   1750#  1750
          1752#  1752   1754#  1754   1756#  1756   1758#  1758   1760#  1760   1762#  1762   1764#  1764
          1766#  1766   1768#  1768   1770#  1770   1772#  1772   1775#  1775   1777#  1777   1870#  1870
          1872#  1872   1874#  1874   1876#  1876   1878#  1878   1880#  1880   1882#  1882   1884#  1884
          1886#  1886   1888#  1888   1890#  1890   1892#  1892   1894#  1894   1896#  1896   1898#  1898
          1900#  1900   1902#  1902   1904#  1904   1906#  1906   1908#  1908   1910#  1910   1912#  1912
          1914#  1914   1916#  1916   1918#  1918   1920#  1920   1922#  1922   1924#  1924   1926#  1926
          1928#  1928   1930#  1930   1932#  1932   1934#  1934   1936#  1936   1938#  1938   1940#  1940
          1942#  1942   1944#  1944   1946#  1946   1948#  1948   1950#  1950   1952#  1952   1954#  1954
          1956#  1956   1958#  1958   1960#  1960   1962#  1962   1964#  1964   1966#  1966   1968#  1968
          1970#  1970   1973#  1973   2045#  2045   2047#  2047   2050#  2050   2052#  2052   2054#  2054
          2056#  2056   2058#  2058   2061#  2061   2063#  2063   2065#  2065   2067#  2067   2070#  2070
          2072#  2072   2074#  2074   2076#  2076   2079#  2079   2081#  2081   2083#  2083   2085#  2085
          2088#  2088   2090#  2090   2092#  2092   2094#  2094   2097#  2097   2099#  2099   2101#  2101
          2103#  2103   2106#  2106   2108#  2108   2110#  2110   2112#  2112   2115#  2115   2117#  2117
          2119#  2119   2121#  2121   2124#  2124   2126#  2126   2128#  2128   2130#  2130   2133#  2133
          2135#  2135   2137#  2137   2139#  2139   2142#  2142   2144#  2144   2146#  2146   2148#  2148
          2151#  2151   2153#  2153   2155#  2155   2157#  2157   2160#  2160   2162#  2162   2164#  2164
          2166#  2166   2169#  2169   2171#  2171   2173#  2173   2175#  2175   2178#  2178   2180#  2180
          2182#  2182   2184#  2184   2187#  2187   2189#  2189   2191#  2191   2193#  2193   2196#  2196
          2198#  2198   2200#  2200   2202#  2202   2204#  2204   2207#  2207   2209#  2209   2211#  2211
          2213#  2213   2215#  2215   2218#  2218   2220#  2220   2222#  2222   2224#  2224   2226#  2226
          2229#  2229   2231#  2231   2233#  2233   2235#  2235   2237#  2237   2240#  2240   2242#  2242
          2244#  2244   2246#  2246   2248#  2248   2251#  2251   2253#  2253   2255#  2255   2257#  2257
          2259#  2259   2262#  2262   2264#  2264   2266#  2266   2268#  2268   2270#  2270   2273#  2273
          2275#  2275   2277#  2277   2279#  2279   2281#  2281   2284#  2284   2286#  2286   2288#  2288
          2290#  2290   2292#  2292   2295#  2295   2297#  2297   2299#  2299   2301#  2301   2303#  2303
          2306#  2306   2308#  2308   2310#  2310   2312#  2312   2314#  2314   2317#  2317   2319#  2319
          2321#  2321   2323#  2323   2325#  2325   2328#  2328   2330#  2330   2332#  2332   2334#  2334
          2336#  2336   2339#  2339   2341#  2341   2343#  2343   2345#  2345   2347#  2347   2350#  2350
          2352#  2352   2354#  2354   2356#  2356   2358#  2358   2361#  2361   2363#  2363   2365#  2365
          2367#  2367   2369#  2369   2374#  2374   2379#  2379
.LA        184    186    597    603    609    615    624    630    636    642    651    657    663    669
           678    684    690    696    705    711    717    723    732    738    744    750    759    765
           771    777    786    792    798    804    813    819    825    831    840    846    852    858
           867    873    879    885    894    900    906    912    921    927    933    939    948    954
           960    966    975    981    987    993   1002   1008   1014   1020   1119   1123   1127   1131
          1135   1139   1143   1147   1151   1155   1159   1163   1167   1171   1175   1179   1184   1188
          1192   1196   1200   1204   1208   1212   1216   1220   1224   1228   1232   1236   1240   1244
          1249   1253   1257   1261   1265   1269   1273   1277   1281   1285   1289   1293   1297   1301
          1305   1309   1314   1318   1322   1326   1330   1334   1338   1342   1346   1350   1354   1358
          1362   1366   1370   1374   1377   1479   1485   1491   1497   1503   1509   1515   1521   1527
```

```
SEQ 1352  1533  1539  1545  1551  1557  1563  1569  1571  1576  1680  1770  1775  1876  1882   SEQ 1365
          1888  1894  1900  1906  1912  1918  1924  1930  1936  1942  1948  1954  1960  1966
          1968  1973  2047  2050  2052  2054  2056  2061  2063  2065  2070  2072  2074  2079
          2081  2083  2088  2090  2092  2097  2099  2101  2106  2108  2110  2115  2117  2119
          2124  2126  2128  2133  2135  2137  2142  2144  2146  2151  2153  2155  2160  2162
          2164  2169  2171  2173  2178  2180  2182  2187  2189  2191  2196  2198  2200  2202
          2207  2209  2211  2213  2218  2220  2222  2224  2229  2231  2233  2235  2240  2242
          2244  2246  2251  2253  2255  2257  2262  2264  2266  2268  2273  2275  2277  2279
          2284  2286  2288  2290  2295  2297  2299  2301  2306  2308  2310  2312  2317  2319
          2321  2323  2328  2330  2332  2334  2339  2341  2343  2345  2350  2352  2354  2356
          2361  2363  2365  2367
.LADDR     170   288   593   620   647   674   701   728   755   782   809   836   863   890
           917   944   971   998  1115  1473  1672  1870  2045  2050  2196  2374  2379
.LAND      170   288   593   620   647   674   701   728   755   782   809   836   863   890
           917   944   971   998  1117  1121  1125  1129  1133  1137  1141  1145  1149  1153
          1157  1161  1165  1169  1173  1177  1182  1186  1190  1194  1198  1202  1206  1210
          1214  1218  1222  1226  1230  1234  1238  1242  1247  1251  1255  1259  1263  1267
          1271  1275  1279  1283  1287  1291  1295  1299  1303  1307  1312  1316  1320  1324
          1328  1332  1336  1340  1344  1348  1352  1356  1360  1364  1368  1372  1477  1483
          1489  1495  1501  1507  1513  1519  1525  1531  1537  1543  1549  1555  1561  1567
          1676  1682  1688  1694  1700  1706  1712  1718  1724  1730  1736  1742  1748  1754
          1760  1766  1874  1880  1886  1892  1898  1904  1910  1916  1922  1928  1934  1940
          1946  1952  1958  1964  2047  2050  2054  2063  2072  2081  2090  2099  2108  2117
          2126  2135  2144  2153  2162  2171  2180  2189  2196  2200  2207  2211  2218  2222
          2229  2233  2240  2244  2251  2255  2262  2266  2273  2277  2284  2288  2295  2299
          2306  2310  2317  2321  2328  2332  2339  2343  2350  2354  2361  2365
.LB        177   191   593   595   599   601   605   607   611   613   620   622   626   628
           632   634   638   640   647   649   653   655   659   661   665   667   674   676
           680   682   686   688   692   694   701   703   707   709   713   715   719   721
           728   730   734   736   740   742   746   748   755   757   761   763   767   769
           773   775   782   784   788   790   794   796   800   802   809   811   815   817
           821   823   827   829   836   838   842   844   848   850   854   856   863   865
           869   871   875   877   881   883   890   892   896   898   902   904   908   910
           917   919   923   925   929   931   935   937   944   946   950   952   956   958
           962   964   971   973   977   979   983   985   989   991   998  1000  1004  1006
          1010  1012  1016  1018  1117  1119  1121  1123  1125  1127  1129  1131  1133  1135
          1137  1139  1141  1143  1145  1147  1149  1151  1153  1155  1157  1159  1161  1163
          1165  1167  1169  1171  1173  1175  1177  1179  1182  1184  1186  1188  1190  1192
          1194  1196  1198  1200  1202  1204  1206  1208  1210  1212  1214  1216  1218  1220
          1222  1224  1226  1228  1230  1232  1234  1236  1238  1240  1242  1244  1247  1249
          1251  1253  1255  1257  1259  1261  1263  1265  1267  1269  1271  1273  1275  1277
          1279  1281  1283  1285  1287  1289  1291  1293  1295  1297  1299  1301  1303  1305
          1307  1309  1312  1314  1316  1318  1320  1322  1324  1326  1328  1330  1332  1334
          1336  1338  1340  1342  1344  1346  1348  1350  1352  1354  1356  1358  1360  1362
          1364  1366  1368  1370  1372  1374  1477  1481  1483  1487  1489  1493  1495  1499
          1501  1505  1507  1511  1513  1517  1519  1523  1525  1529  1531  1535  1537  1541
          1543  1547  1549  1553  1555  1559  1561  1565  1567  1571  1676  1678  1680  1682
          1684  1686  1688  1690  1692  1694  1696  1698  1700  1702  1704  1706  1708  1710
          1712  1714  1716  1718  1720  1722  1724  1726  1728  1730  1732  1734  1736  1738
          1740  1742  1744  1746  1748  1750  1752  1754  1756  1758  1760  1762  1764  1766
          1768  1770  1772  1874  1876  1878  1880  1882  1884  1886  1888  1890  1892  1894
          1896  1898  1900  1902  1904  1906  1908  1910  1912  1914  1916  1918  1920  1922
          1924  1926  1928  1930  1932  1934  1936  1938  1940  1942  1944  1946  1948  1950
```

```
                1952   1954   1956   1958   1960   1962   1964   1966   1968   1970   2047   2050   2052   2054      SEQ 1366
                2061   2063   2070   2072   2079   2081   2088   2090   2097   2099   2106   2108   2115   2117
                2124   2126   2133   2135   2142   2144   2151   2153   2160   2162   2169   2171   2178   2180
                2187   2189   2196   2198   2200   2207   2209   2211   2218   2220   2222   2229   2231   2233
                2240   2242   2244   2251   2253   2255   2262   2264   2266   2273   2275   2277   2284   2286
                2288   2295   2297   2299   2306   2308   2310   2317   2319   2321   2328   2330   2332   2339
                2341   2343   2350   2352   2354   2361   2363   2365
.LBAD           2374   2379
.LCCFZ          2056   2065   2074   2083   2092   2101   2110   2119   2128   2137   2146   2155   2164   2173
                2182   2191   2202   2213   2224   2235   2246   2257   2268   2279   2290   2301   2312   2323
                2334   2345   2356   2367
.LCENA          2056   2065   2074   2083   2092   2101   2110   2119   2128   2137   2146   2155   2164   2173
                2182   2191   2202   2213   2224   2235   2246   2257   2268   2279   2290   2301   2312   2323
                2334   2345   2356   2367
.LCJP           2056   2065   2074   2083   2092   2101   2110   2119   2128   2137   2146   2155   2164   2173
                2182   2191   2202   2213   2224   2235   2246   2257   2268   2279   2290   2301   2312   2323
                2334   2345   2356   2367
.LCONT           172    179    186    193    290    297    299    301    303    312    314    316    318    327
                 329    331    333    342    344    346    348    595    601    607    613    622    628    634
                 640    649    655    661    667    676    682    688    694    703    709    715    721    730
                 736    742    748    757    763    769    775    784    790    796    802    811    817    823
                 829    838    844    850    856    865    871    877    883    892    898    904    910    919
                 925    931    937    946    952    958    964    973    979    985    991   1000   1006   1012
                1018   1117   1119   1121   1123   1125   1127   1129   1131   1133   1135   1137   1139   1141
                1143   1145   1147   1149   1151   1153   1155   1157   1159   1161   1163   1165   1167   1169
                1171   1173   1175   1177   1179   1182   1184   1186   1188   1190   1192   1194   1196   1198
                1200   1202   1204   1206   1208   1210   1212   1214   1216   1218   1220   1222   1224   1226
                1228   1230   1232   1234   1236   1238   1240   1242   1244   1247   1249   1251   1253   1255
                1257   1259   1261   1263   1265   1267   1269   1271   1273   1275   1277   1279   1281   1283
                1285   1287   1289   1291   1293   1295   1297   1299   1301   1303   1305   1307   1309   1312
                1314   1316   1318   1320   1322   1324   1326   1328   1330   1332   1334   1336   1338   1340
                1342   1344   1346   1348   1350   1352   1354   1356   1358   1360   1362   1364   1366   1368
                1370   1372   1374   1477   1479   1481   1483   1485   1487   1489   1491   1493   1495   1497
                1499   1501   1503   1505   1507   1509   1511   1513   1515   1517   1519   1521   1523   1525
                1527   1529   1531   1533   1535   1537   1539   1541   1543   1545   1547   1549   1551   1553
                1555   1557   1559   1561   1563   1565   1567   1569   1571   1676   1678   1680   1682   1684
                1686   1688   1690   1692   1694   1696   1698   1700   1702   1704   1706   1708   1710   1712
                1714   1716   1718   1720   1722   1724   1726   1728   1730   1732   1734   1736   1738   1740
                1742   1744   1746   1748   1750   1752   1754   1756   1758   1760   1762   1764   1766   1768
                1770   1874   1876   1878   1880   1882   1884   1886   1888   1890   1892   1894   1896   1898
                1900   1902   1904   1906   1908   1910   1912   1914   1916   1918   1920   1922   1924   1926
                1928   1930   1932   1934   1936   1938   1940   1942   1944   1946   1948   1950   1952   1954
                1956   1958   1960   1962   1964   1966   1968   2052   2054   2061   2063   2070   2072   2079
                2081   2088   2090   2097   2099   2106   2108   2115   2117   2124   2126   2133   2135   2142
                2144   2151   2153   2160   2162   2169   2171   2178   2180   2187   2189   2196   2198   2200
                2207   2209   2211   2218   2220   2222   2229   2231   2233   2240   2242   2244   2251   2253
                2255   2262   2264   2266   2273   2275   2277   2284   2286   2288   2295   2297   2299   2306
                2308   2310   2317   2319   2321   2328   2330   2332   2339   2341   2343   2350   2352   2354
                2361   2363   2365
.LCRY            172    290    595    622    649    676    703    730    757    784    811    838    865    892
                 919    946    973   1000
.LD              170    172    174    177    179    181    184    186    188    191    193    195    288    290
                 295    297    299    301    303    305    310    312    314    316    318    320    325    327
```

```
         329   331   333   335   340   342   344   346   348   350   353   593   595   597    SEQ 1367
         599   601   603   605   607   609   611   613   615   620   622   624   626   628
         630   632   634   636   638   640   642   647   649   651   653   655   657   659
         661   663   665   667   669   674   676   678   680   682   684   686   688   690
         692   694   696   701   703   705   707   709   711   713   715   717   719   721
         723   728   730   732   734   736   738   740   742   744   746   748   750   755
         757   759   761   763   765   767   769   771   773   775   777   782   784   786
         788   790   792   794   796   798   800   802   804   809   811   813   815   817
         819   821   823   825   827   829   831   836   838   840   842   844   846   848
         850   852   854   856   858   863   865   867   869   871   873   875   877   879
         881   883   885   890   892   894   896   898   900   902   904   906   908   910
         912   917   919   921   923   925   927   929   931   933   935   937   939   944
         946   948   950   952   954   956   958   960   962   964   966   971   973   975
         977   979   981   983   985   987   989   991   993   998  1000  1002  1004  1006
        1008  1010  1012  1014  1016  1018  1020  1115  1117  1119  1121  1123  1125  1127
        1129  1131  1133  1135  1137  1139  1141  1143  1145  1147  1149  1151  1153  1155
        1157  1159  1161  1163  1165  1167  1169  1171  1173  1175  1177  1179  1182  1184
        1186  1188  1190  1192  1194  1196  1198  1200  1202  1204  1206  1208  1210  1212
        1214  1216  1218  1220  1222  1224  1226  1228  1230  1232  1234  1236  1238  1240
        1242  1244  1247  1249  1251  1253  1255  1257  1259  1261  1263  1265  1267  1269
        1271  1273  1275  1277  1279  1281  1283  1285  1287  1289  1291  1293  1295  1297
        1299  1301  1303  1305  1307  1309  1312  1314  1316  1318  1320  1322  1324  1326
        1328  1330  1332  1334  1336  1338  1340  1342  1344  1346  1348  1350  1352  1354
        1356  1358  1360  1362  1364  1366  1368  1370  1372  1374  1473  1475  1477  1479
        1481  1483  1485  1487  1489  1491  1493  1495  1497  1499  1501  1503  1505  1507
        1509  1511  1513  1515  1517  1519  1521  1523  1525  1527  1529  1531  1533  1535
        1537  1539  1541  1543  1545  1547  1549  1551  1553  1555  1557  1559  1561  1563
        1565  1567  1569  1571  1573  1672  1674  1676  1678  1680  1682  1684  1686  1688
        1690  1692  1694  1696  1698  1700  1702  1704  1706  1708  1710  1712  1714  1716
        1718  1720  1722  1724  1726  1728  1730  1732  1734  1736  1738  1740  1742  1744
        1746  1748  1750  1752  1754  1756  1758  1760  1762  1764  1766  1768  1770  1772
        1870  1872  1874  1876  1878  1880  1882  1884  1886  1888  1890  1892  1894  1896
        1898  1900  1902  1904  1906  1908  1910  1912  1914  1916  1918  1920  1922  1924
        1926  1928  1930  1932  1934  1936  1938  1940  1942  1944  1946  1948  1950  1952
        1954  1956  1958  1960  1962  1964  1966  1968  1970  1973  2047  2050  2052  2054
        2061  2063  2070  2072  2079  2081  2088  2090  2097  2099  2106  2108  2115  2117
        2124  2126  2133  2135  2142  2144  2151  2153  2160  2162  2169  2171  2178  2180
        2187  2189  2196  2198  2200  2207  2209  2211  2218  2220  2222  2229  2231  2233
        2240  2242  2244  2251  2253  2255  2262  2264  2266  2273  2275  2277  2284  2286
        2288  2295  2297  2299  2306  2308  2310  2317  2319  2321  2328  2330  2332  2339
        2341  2343  2350  2352  2354  2361  2363  2365
.LJ      170   174   177   181   184   188   191   195   288   295   305   310   320   325
         335   340   350   353   593   597   599   603   605   609   611   615   620   624
         626   630   632   636   638   642   647   651   653   657   659   663   665   669
         674   678   680   684   686   690   692   696   701   705   707   711   713   717
         719   723   728   732   734   738   740   744   746   750   755   759   761   765
         767   771   773   777   782   786   788   792   794   798   800   804   809   813
         815   819   821   825   827   831   836   840   842   846   848   852   854   858
         863   867   869   873   875   879   881   885   890   894   896   900   902   906
         908   912   917   921   923   927   929   933   935   939   944   948 [illegible] 954
         956   960   962   966   971   975   977   981   983   987   989   993 [illegible] 1002
        1004  1008  1010  1014  1016  1020  1115  1377  1379  1473  1475  1573  1576  1578
        1672  1674  1772  1775  1777  1870  1872  1970  1973  2045  2050  2056  2058  2065
```

```
1355
            2067  2074  2076  2083  2085  2092  2094  2101  2103  2110  2112  2119  2121  2128    SEQ 1368
            2130  2137  2139  2146  2148  2155  2157  2164  2166  2173  2175  2182  2184  2191
            2193  2202  2204  2213  2215  2224  2226  2235  2237  2246  2248  2257  2259  2268
            2270  2279  2281  2290  2292  2301  2303  2312  2314  2323  2325  2334  2336  2345
            2347  2356  2358  2367  2369  2374  2379
.LJMAP       170   174   177   181   184   188   191   195   288   353   593   597   599   603
             605   609   611   615   620   624   626   630   632   636   638   642   647   651
             653   657   659   663   665   669   674   678   680   684   686   690   692   696
             701   705   707   711   713   717   719   723   728   732   734   738   740   744
             746   750   755   759   761   765   767   771   773   777   782   786   788   792
             794   798   800   804   809   813   815   819   821   825   827   831   836   840
             842   846   848   852   854   858   863   867   869   873   875   879   881   885
             890   894   896   900   902   906   908   912   917   921   923   927   929   933
             935   939   944   948   950   954   956   960   962   966   971   975   977   981
             983   987   989   993   998  1002  1004  1008  1010  1014  1016  1020  1115  1377
            1379  1473  1576  1578  1672  1775  1777  1870  1973  2045  2050  2058  2067  2076
            2085  2094  2103  2112  2121  2130  2139  2148  2157  2166  2175  2184  2193  2204
            2215  2226  2237  2248  2259  2270  2281  2292  2303  2314  2325  2336  2347  2358
            2369  2374  2379
.LJZ        2047
.LLDCT       295   310   325   340  1475  1674  1872
.LMGC        174   181   188   195   353   597   603   609   615   624   630   636   642   651
             657   663   669   678   684   690   696   705   711   717   723   732   738   744
             750   759   765   771   777   786   792   798   804   813   819   825   831   840
             846   852   858   867   873   879   885   894   900   906   912   921   927   933
             939   948   954   960   966   975   981   987   993  1002  1008  1014  1020  1115
            1377  1473  1576  1672  1775  1870  1973  2054  2063  2072  2081  2090  2099  2108
            2117  2126  2135  2144  2153  2162  2171  2180  2189  2200  2211  2222  2233  2244
            2255  2266  2277  2288  2299  2310  2321  2332  2343  2354  2365
.LNAND      1314  1318  1322  1326  1330  1334  1338  1342  1346  1350  1354  1358  1362  1366
            1370  1374  1497  1499  1521  1523  1545  1547  1569  1571  1696  1698  1720  1722
            1744  1746  1768  1770  1894  1896  1918  1920  1942  1944  1966  1968
.LOENA       174   181   188   195   353   597   603   609   615   624   630   636   642   651
             657   663   669   678   684   690   696   705   711   717   723   732   738   744
             750   759   765   771   777   786   792   798   804   813   819   825   831   840
             846   852   858   867   873   879   885   894   900   906   912   921   927   933
             939   948   954   960   966   975   981   987   993  1002  1008  1014  1020  1377
            1576  1775  1973
.LOR         174   177   179   181   184   186   188   191   193   195   297   299   301   303
             305   312   314   316   318   320   327   329   331   333   335   342   344   346
             348   350   353   597   599   601   603   605   607   609   611   613   615   624
             626   628   630   632   634   636   638   640   642   651   653   655   657   659
             661   663   665   667   669   678   680   682   684   686   688   690   692   694
             696   705   707   709   711   713   715   717   719   721   723   732   734   736
             738   740   742   744   746   748   750   759   761   763   765   767   769   771
             773   775   777   786   788   790   792   794   796   798   800   802   804   813
             815   817   819   821   823   825   827   829   831   840   842   844   846   848
             850   852   854   856   858   867   869   871   873   875   877   879   881   883
             885   894   896   898   900   902   904   906   908   910   912   921   923   925
             927   929   931   933   935   937   939   948   950   952   954   956   958   960
             962   964   966   975   977   979   981   983   985   987   989   991   993  1002
            1004  1006  1008  1010  1012  1014  1016  1018  1020  1115  1184  1188  1192  1196
            1200  1204  1208  1212  1216  1220  1224  1228  1232  1236  1240  1244  1377  1473
```

```
.RSD
.RSD
.RSE
.RSK
.RXN
.RXO
```

```
            1485  1487  1509  1511  1533  1535  1557  1559  1576  1672  1684  1686  1708  1710          ATAB
```

```
             1485   1487   1509   1511   1533   1535   1557   1559   1576   1672   1684   1686   1708   1710
             1732   1734   1756   1758   1772   1775   1870   1882   1884   1906   1908   1930   1932   1954
             1956   1970   1973   2056   2065   2074   2083   2092   2101   2110   2119   2128   2137   2146
             2155   2164   2173   2182   2191   2202   2213   2224   2235   2246   2257   2268   2279   2290
             2301   2312   2323   2334   2345   2356   2367
.LPLUS        172    290    595    622    649    676    703    730    757    784    811    838    865    892
              919    946    973   1000   1119   1123   1127   1131   1135   1139   1143   1147   1151   1155
             1159   1163   1167   1171   1175   1179   1479   1481   1503   1505   1527   1529   1551   1553
             1678   1680   1702   1704   1726   1728   1750   1752   1876   1878   1900   1902   1924   1926
             1948   1950
.LRPCT        305    320    335    350   1573   1772   1970
.LSOA         170    184    186    288    597    603    609    615    624    630    636    642    651    657
              663    669    678    684    690    696    705    711    717    723    732    738    744    750
              759    765    771    777    786    792    798    804    813    819    825    831    840    846
              852    858    867    873    879    885    894    900    906    912    921    927    933    939
              948    954    960    966    975    981    987    993   1002   1008   1014   1020   1117   1119
             1121   1123   1125   1127   1129   1131   1133   1135   1137   1139   1141   1143   1145   1147
             1149   1151   1153   1155   1157   1159   1161   1163   1165   1167   1169   1171   1173   1175
             1177   1179   1182   1184   1186   1188   1190   1192   1194   1196   1198   1200   1202   1204
             1206   1208   1210   1212   1214   1216   1218   1220   1222   1224   1226   1228   1230   1232
             1234   1236   1238   1240   1242   1244   1247   1249   1251   1253   1255   1257   1259   1261
             1263   1265   1267   1269   1271   1273   1275   1277   1279   1281   1283   1285   1287   1289
             1291   1293   1295   1297   1299   1301   1303   1305   1307   1309   1312   1314   1316   1318
             1320   1322   1324   1326   1328   1330   1332   1334   1336   1338   1340   1342   1344   1346
             1348   1350   1352   1354   1356   1358   1360   1362   1364   1366   1368   1370   1372   1374
             1377   1477   1479   1483   1485   1489   1491   1495   1497   1501   1503   1507   1509   1513
             1515   1519   1521   1525   1527   1531   1533   1537   1539   1543   1545   1549   1551   1555
             1557   1561   1563   1567   1569   1576   1676   1680   1682   1686   1688   1692   1694   1698
             1700   1704   1706   1710   1712   1716   1718   1722   1724   1728   1730   1734   1736   1740
             1742   1746   1748   1752   1754   1758   1760   1764   1766   1770   1775   1874   1876   1880
             1882   1886   1888   1892   1894   1898   1900   1904   1906   1910   1912   1916   1918   1922
             1924   1928   1930   1934   1936   1940   1942   1946   1948   1952   1954   1958   1960   1964
             1966   1973   2047   2050   2052   2056   2061   2065   2070   2074   2079   2083   2088   2092
             2097   2101   2106   2110   2115   2119   2124   2128   2133   2137   2142   2146   2151   2155
             2160   2164   2169   2173   2178   2182   2187   2191   2196   2198   2202   2207   2209   2213
             2218   2220   2224   2229   2231   2235   2240   2242   2246   2251   2253   2257   2262   2264
             2268   2273   2275   2279   2284   2286   2290   2295   2297   2301   2306   2308   2312   2317
             2319   2323   2328   2330   2334   2339   2341   2345   2350   2352   2356   2361   2363   2367
.LSOB         593    595    599    601    611    613    620    622    626    628    638    640    647    649
              653    655    665    667    674    676    680    682    692    694    701    703    707    709
              719    721    728    730    734    736    746    748    755    757    761    763    773    775
              782    784    788    790    800    802    809    811    815    817    827    829    836    838
              842    844    854    856    863    865    869    871    881    883    890    892    896    898
              908    910    917    919    923    925    935    937    944    946    950    952    962    964
              971    973    977    979    989    991    998   1000   1004   1006   1016   1018   1678   1684
             1690   1696   1702   1708   1714   1720   1726   1732   1738   1744   1750   1756   1762   1768
             1772   1970
.LSOQ         172    174    177    179    181    188    191    193    195    290    297    299    301    303
              305    312    314    316    318    320    327    329    331    333    335    342    344    346
              348    350    353    605    607    632    634    659    661    686    688    713    715    740
              742    767    769    794    796    821    823    848    850    875    877    902    904    929
              931    956    958    983    985   1010   1012   1481   1487   1493   1499   1505   1511   1517
             1523   1529   1535   1541   1547   1553   1559   1565   1571   1878   1884   1890   1896   1902
```

ATAB
CALC

CON

```
             1908   1914   1920   1926   1932   1938   1944   1950   1956   1962   1968
```

```
        1908 1914 1920 1926 1932 1938 1944 1950 1956 1962 1968
.LSDO   1115 1473 1672 1870
.LSDA   2054 2063 2072 2081 2090 2099 2108 2117 2126 2135 2144 2153 2162 2171
        2180 2189 2200 2211 2222 2233 2244 2255 2266 2277 2288 2299 2310 2321
        2332 2343 2354 2365
.LSELE   174  181  188  195  353  597  603  609  615  624  630  636  642  651
         657  663  669  678  684  690  696  705  711  717  723  732  738  744
         750  759  765  771  777  786  792  798  804  813  819  825  831  840
         846  852  858  867  873  879  885  894  900  906  912  921  927  933
         939  948  954  960  966  975  981  987  993 1002 1008 1014 1020 1377
        1576 1775 1973
.LSKCN  1115 1473 1672 1870 2054 2063 2072 2081 2090 2099 2108 2117 2126 2135
        2144 2153 2162 2171 2180 2189 2200 2211 2222 2233 2244 2255 2266 2277
        2288 2299 2310 2321 2332 2343 2354 2365
.LXNOR  2052 2061 2070 2079 2088 2097 2106 2115 2124 2133 2142 2151 2160 2169
        2178 2187 2198 2209 2220 2231 2242 2253 2264 2275 2286 2297 2308 2319
        2330 2341 2352 2363
.LXOR   1249 1253 1257 1261 1265 1269 1273 1277 1281 1285 1289 1293 1297 1301
        1305 1309 1491 1493 1515 1517 1539 1541 1563 1565 1690 1692 1714 1716
        1738 1740 1762 1764 1888 1890 1912 1914 1936 1938 1960 1962
.MA      184  186  597  603  609  615  624  630  636  642  651  657  663  669
         678  684  690  696  705  711  717  723  732  738  744  750  759  765
         771  777  786  792  798  804  813  819  825  831  840  846  852  858
         867  873  879  885  894  900  906  912  921  927  933  939  948  954
         960  966  975  981  987  993 1002 1008 1014 1020 1119 1123 1127 1131
        1135 1139 1143 1147 1151 1155 1159 1163 1167 1171 1175 1179 1184 1188
        1192 1196 1200 1204 1208 1212 1216 1220 1224 1228 1232 1236 1240 1244
        1249 1253 1257 1261 1265 1269 1273 1277 1281 1285 1289 1293 1297 1301
        1305 1309 1314 1318 1322 1326 1330 1334 1338 1342 1346 1350 1354 1358
        1362 1366 1370 1374 1377 1479 1485 1491 1497 1503 1509 1515 1521 1527
        1533 1539 1545 1551 1557 1563 1569 1571 1576 1680 1770 1775 1876 1882
        1888 1894 1900 1906 1912 1918 1924 1930 1936 1942 1948 1954 1960 1966
        1968 1973 2047 2050 2052 2054 2056 2061 2063 2065 2070 2072 2074 2079
        2081 2083 2088 2090 2092 2097 2099 2101 2106 2108 2110 2115 2117 2119
        2124 2126 2128 2133 2135 2137 2142 2144 2146 2151 2153 2155 2160 2162
        2164 2169 2171 2173 2178 2180 2182 2187 2189 2191 2196 2198 2200 2202
        2207 2209 2211 2213 2218 2220 2222 2224 2229 2231 2233 2235 2240 2242
        2244 2246 2251 2253 2255 2257 2262 2264 2266 2268 2273 2275 2277 2279
        2284 2286 2288 2290 2295 2297 2299 2301 2306 2308 2310 2312 2317 2319
        2321 2323 2328 2330 2332 2334 2339 2341 2343 2345 2350 2352 2354 2356
        2361 2363 2365 2367
.MAND    170  288  593  620  647  674  701  728  755  782  809  836  863  890
         917  944  971  998 1117 1121 1125 1129 1133 1137 1141 1145 1149 1153
        1157 1161 1165 1169 1173 1177 1182 1186 1190 1194 1198 1202 1206 1210
        1214 1218 1222 1226 1230 1234 1238 1242 1247 1251 1255 1259 1263 1267
        1271 1275 1279 1283 1287 1291 1295 1299 1303 1307 1312 1316 1320 1324
        1328 1332 1336 1340 1344 1348 1352 1356 1360 1364 1368 1372 1477 1483
        1489 1495 1501 1507 1513 1519 1525 1531 1537 1543 1549 1555 1561 1567
        1676 1682 1688 1694 1700 1706 1712 1718 1724 1730 1736 1742 1748 1754
        1760 1766 1874 1880 1886 1892 1898 1904 1910 1916 1922 1928 1934 1940
        1946 1952 1958 1964 2047 2050 2054 2063 2072 2081 2090 2099 2108 2117
        2126 2135 2144 2153 2162 2171 2180 2189 2196 2200 2207 2211 2218 2222
        2229 2233 2240 2244 2251 2255 2262 2266 2273 2277 2284 2288 2295 2299
```

FIE

```
           2306   2310   2317   2321   2328   2332   2339   2343   2350   2354   2361   2365                  SEQ 1371
.MB         177    191    593    595    599    601    605    607    611    613    620    622    626    628
            632    634    638    640    647    649    653    655    659    661    665    667    674    676
            680    682    686    688    692    694    701    703    707    709    713    715    719    721
            728    730    734    736    740    742    746    748    755    757    761    763    767    769
            773    775    782    784    788    790    794    796    800    802    809    811    815    817
            821    823    827    829    836    838    842    844    848    850    854    856    863    865
            869    871    875    877    881    883    890    892    896    898    902    904    908    910
            917    919    923    925    929    931    935    937    944    946    950    952    956    958
            962    964    971    973    977    979    983    985    989    991    998   1000   1004   1006
           1010   1012   1016   1018   1117   1119   1121   1123   1125   1127   1129   1131   1133   1135
           1137   1139   1141   1143   1145   1147   1149   1151   1153   1155   1157   1159   1161   1163
           1165   1167   1169   1171   1173   1175   1177   1179   1182   1184   1186   1188   1190   1192
           1194   1196   1198   1200   1202   1204   1206   1208   1210   1212   1214   1216   1218   1220
           1222   1224   1226   1228   1230   1232   1234   1236   1238   1240   1242   1244   1247   1249
           1251   1253   1255   1257   1259   1261   1263   1265   1267   1269   1271   1273   1275   1277
           1279   1281   1283   1285   1287   1289   1291   1293   1295   1297   1299   1301   1303   1305
           1307   1309   1312   1314   1316   1318   1320   1322   1324   1326   1328   1330   1332   1334
           1336   1338   1340   1342   1344   1346   1348   1350   1352   1354   1356   1358   1360   1362
           1364   1366   1368   1370   1372   1374   1477   1481   1483   1487   1489   1493   1495   1499
           1501   1505   1507   1511   1513   1517   1519   1523   1525   1529   1531   1535   1537   1541
           1543   1547   1549   1553   1555   1559   1561   1565   1567   1571   1676   1678   1680   1682
           1684   1686   1688   1690   1692   1694   1696   1698   1700   1702   1704   1706   1708   1710
           1712   1714   1716   1718   1720   1722   1724   1726   1728   1730   1732   1734   1736   1738
           1740   1742   1744   1746   1748   1750   1752   1754   1756   1758   1760   1762   1764   1766
           1768   1770   1772   1874   1876   1878   1880   1882   1884   1886   1888   1890   1892   1894
           1896   1898   1900   1902   1904   1906   1908   1910   1912   1914   1916   1918   1920   1922
           1924   1926   1928   1930   1932   1934   1936   1938   1940   1942   1944   1946   1948   1950
           1952   1954   1956   1958   1960   1962   1964   1966   1968   1970   2047   2050   2052   2054
           2061   2063   2070   2072   2079   2081   2088   2090   2097   2099   2106   2108   2115   2117
           2124   2126   2133   2135   2142   2144   2151   2153   2160   2162   2169   2171   2178   2180
           2187   2189   2196   2198   2200   2207   2209   2211   2218   2220   2222   2229   2231   2233
           2240   2242   2244   2251   2253   2255   2262   2264   2266   2273   2275   2277   2284   2286
           2288   2295   2297   2299   2306   2308   2310   2317   2319   2321   2328   2330   2332   2339
           2341   2343   2350   2352   2354   2361   2363   2365
.MBAD      2374   2379
.MCCFZ     2056   2065   2074   2083   2092   2101   2110   2119   2128   2137   2146   2155   2164   2173
           2182   2191   2202   2213   2224   2235   2246   2257   2268   2279   2290   2301   2312   2323
           2334   2345   2356   2367
.MCENA     2056   2065   2074   2083   2092   2101   2110   2119   2128   2137   2146   2155   2164   2173
           2182   2191   2202   2213   2224   2235   2246   2257   2268   2279   2290   2301   2312   2323
           2334   2345   2356   2367
.MCJP      2056   2065   2074   2083   2092   2101   2110   2119   2128   2137   2146   2155   2164   2173
           2182   2191   2202   2213   2224   2235   2246   2257   2268   2279   2290   2301   2312   2323
           2334   2345   2356   2367
.MCONT      172    179    186    193    290    297    299    301    303    312    314    316    318    327
            329    331    333    342    344    346    348    595    601    607    613    622    628    634
            640    649    655    661    667    676    682    688    694    703    709    715    721    730
            736    742    748    757    763    769    775    784    790    796    802    811    817    823
            829    838    844    850    856    865    871    877    883    892    898    904    910    919
            925    931    937    946    952    958    964    973    979    985    991   1000   1006   1012
           1018   1117   1119   1121   1123   1125   1127   1129   1131   1133   1135   1137   1139   1141
           1143   1145   1147   1149   1151   1153   1155   1157   1159   1161   1163   1165   1167   1169
           1171   1173   1175   1177   1179   1182   1184   1186   1188   1190   1192   1194   1196   1198
           1200   1202   1204   1206   1208   1210   1212   1214   1216   1218   1220   1222   1224   1226
           1228   1230   1232   1234   1236   1238   1240   1242   1244   1247   1249   1251   1253   1255
           1257   1259   1261   1263   1265   1267   1269   1271   1273   1275   1277   1279   1281   1283
           1285   1287   1289   1291   1293   1295   1297   1299   1301   1303   1305   1307   1309   1312
           1314   1316   1318   1320   1322   1324   1326   1328   1330   1332   1334   1336   1338   1340
           1342   1344   1346   1348   1350   1352   1354   1356   1358   1360   1362   1364   1366   1368
           1370   1372   1374   1477   1479   1481   1483   1485   1487   1489   1491   1493   1495   1497
           1499   1501   1503   1505   1507   1509   1511   1513   1515   1517   1519   1521   1523   1525
           1527   1529   1531   1533   1535   1537   1539   1541   1543   1545   1547   1549   1551   1553
           1555   1557   1559   1561   1563   1565   1567   1569   1571   1676   1678   1680   1682   1684
           1686   1688   1690   1692   1694   1696   1698   1700   1702   1704   1706   1708   1710   1712
           1714   1716   1718   1720   1722   1724   1726   1728   1730   1732   1734   1736   1738   1740
           1742   1744   1746   1748   1750   1752   1754   1756   1758   1760   1762   1764   1766   1768
           1770   1874   1876   1878   1880   1882   1884   1886   1888   1890   1892   1894   1896   1898
           1900   1902   1904   1906   1908   1910   1912   1914   1916   1918   1920   1922   1924   1926
           1928   1930   1932   1934   1936   1938   1940   1942   1944   1946   1948   1950   1952   1954
           1956   1958   1960   1962   1964   1966   1968   2052   2054   2061   2063   2070   2072   2079
           2081   2088   2090   2097   2099   2106   2108   2115   2117   2124   2126   2133   2135   2142
           2144   2151   2153   2160   2162   2169   2171   2178   2180   2187   2189   2196   2198   2200
           2207   2209   2211   2218   2220   2222   2229   2231   2233   2240   2242   2244   2251   2253
           2255   2262   2264   2266   2273   2275   2277   2284   2286   2288   2295   2297   2299   2306
           2308   2310   2317   2319   2321   2328   2330   2332   2339   2341   2343   2350   2352   2354
           2361   2363   2365
.MCRY       172    290    595    622    649    676    703    730    757    784    811    838    865    892
            919    946    973   1000
.MD         170    172    174    177    179    181    184    186    188    191    193    195    288    290
            295    297    299    301    303    305    310    312    314    316    318    320    325    327
            329    331    333    335    340    342    344    346    348    350    353    593    595    597
            599    601    603    605    607    609    611    613    615    620    622    624    626    628
            630    632    634    636    638    640    642    647    649    651    653    655    657    659
            661    663    665    667    669    674    676    678    680    682    684    686    688    690
            692    694    696    701    703    705    707    709    711    713    715    717    719    721
            723    728    730    732    734    736    738    740    742    744    746    748    750    755
            757    759    761    763    765    767    769    771    773    775    777    782    784    786
            788    790    792    794    796    798    800    802    804    809    811    813    815    817
            819    821    823    825    827    829    831    836    838    840    842    844    846    848
            850    852    854    856    858    863    865    867    869    871    873    875    877    879
            881    883    885    890    892    894    896    898    900    902    904    906    908    910
            912    917    919    921    923    925    927    929    931    933    935    937    939    944
            946    948    950    952    954    956    958    960    962    964    966    971    973    975
            977    979    981    983    985    987    989    991    993    998   1000   1002   1004   1006
           1008   1010   1012   1014   1016   1018   1020   1115   1117   1119   1121   1123   1125   1127
           1129   1131   1133   1135   1137   1139   1141   1143   1145   1147   1149   1151   1153   1155
           1157   1159   1161   1163   1165   1167   1169   1171   1173   1175   1177   1179   1182   1184
           1186   1188   1190   1192   1194   1196   1198   1200   1202   1204   1206   1208   1210   1212
           1214   1216   1218   1220   1222   1224   1226   1228   1230   1232   1234   1236   1238   1240
           1242   1244   1247   1249   1251   1253   1255   1257   1259   1261   1263   1265   1267   1269
           1271   1273   1275   1277   1279   1281   1283   1285   1287   1289   1291   1293   1295   1297
           1299   1301   1303   1305   1307   1309   1312   1314   1316   1318   1320   1322   1324   1326
           1328   1330   1332   1334   1336   1338   1340   1342   1344   1346   1348   1350   1352   1354
           1356   1358   1360   1362   1364   1366   1368   1370   1372   1374   1473   1475   1477   1479
           1481   1483   1485   1487   1489   1491   1493   1495   1497   1499   1501   1503   1505   1507
```

```
         1509  1511  1513  1515  1517  1519  1521  1523  1525  1527  1529  1531  1533  1535
         1537  1539  1541  1543  1545  1547  1549  1551  1553  1555  1557  1559  1561  1563
         1565  1567  1569  1571  1573  1672  1674  1676  1678  1680  1682  1684  1686  1688
         1690  1692  1694  1696  1698  1700  1702  1704  1706  1708  1710  1712  1714  1716
         1718  1720  1722  1724  1726  1728  1730  1732  1734  1736  1738  1740  1742  1744
         1746  1748  1750  1752  1754  1756  1758  1760  1762  1764  1766  1768  1770  1772
         1870  1872  1874  1876  1878  1880  1882  1884  1886  1888  1890  1892  1894  1896
         1898  1900  1902  1904  1906  1908  1910  1912  1914  1916  1918  1920  1922  1924
         1926  1928  1930  1932  1934  1936  1938  1940  1942  1944  1946  1948  1950  1952
         1954  1956  1958  1960  1962  1964  1966  1968  1970  1973  2047  2050  2052  2054
         2061  2063  2070  2072  2079  2081  2088  2090  2097  2099  2106  2108  2115  2117
         2124  2126  2133  2135  2142  2144  2151  2153  2160  2162  2169  2171  2178  2180
         2187  2189  2196  2198  2200  2207  2209  2211  2218  2220  2222  2229  2231  2233
         2240  2242  2244  2251  2253  2255  2262  2264  2266  2273  2275  2277  2284  2286
         2288  2295  2297  2299  2306  2308  2310  2317  2319  2321  2328  2330  2332  2339
         2341  2343  2350  2352  2354  2361  2363  2365
.MJ       170   174   177   181   184   188   191   195   288   295   305   310   320   325
          335   340   350   353   593   597   599   603   605   609   611   615   620   624
          626   630   632   636   638   642   647   651   653   657   659   663   665   669
          674   678   680   684   686   690   692   696   701   705   707   711   713   717
          719   723   728   732   734   738   740   744   746   750   755   759   761   765
          767   771   773   777   782   786   788   792   794   798   800   804   809   813
          815   819   821   825   827   831   836   840   842   846   848   852   854   858
          863   867   869   873   875   879   881   885   890   894   896   900   902   906
          908   912   917   921   923   927   929   933   935   939   944   948   950   954
          956   960   962   966   971   975   977   981   983   987   989   993   998  1002
         1004  1008  1010  1014  1016  1020  1115  1377  1379  1473  1475  1573  1576  1578
         1672  1674  1772  1775  1777  1870  1872  1970  1973  2045  2050  2056  2058  2065
         2067  2074  2076  2083  2085  2092  2094  2101  2103  2110  2112  2119  2121  2128
         2130  2137  2139  2146  2148  2155  2157  2164  2166  2173  2175  2182  2184  2191
         2193  2202  2204  2213  2215  2224  2226  2235  2237  2246  2248  2257  2259  2268
         2270  2279  2281  2290  2292  2301  2303  2312  2314  2323  2325  2334  2336  2345
         2347  2356  2358  2367  2369  2374  2379
.MJMAP    170   174   177   181   184   188   191   195   288   353   593   597   599   603
          605   609   611   615   620   624   626   630   632   636   638   642   647   651
          653   657   659   663   665   669   674   678   680   684   686   690   692   696
          701   705   707   711   713   717   719   723   728   732   734   738   740   744
          746   750   755   759   761   765   767   771   773   777   782   786   788   792
          794   798   800   804   809   813   815   819   821   825   827   831   836   840
          842   846   848   852   854   858   863   867   869   873   875   879   881   885
          890   894   896   900   902   906   908   912   917   921   923   927   929   933
          935   939   944   948   950   954   956   960   962   966   971   975   977   981
          983   987   989   993   998  1002  1004  1008  1010  1014  1016  1020  1115  1377
         1379  1473  1576  1578  1672  1775  1777  1870  1973  2045  2050  2058  2067  2076
         2085  2094  2103  2112  2121  2130  2139  2148  2157  2166  2175  2184  2193  2204
         2215  2226  2237  2248  2259  2270  2281  2292  2303  2314  2325  2336  2347  2358
         2369  2374  2379
.MJZ     2047
.MLDCT    295   310   325   340  1475  1674  1872
.MMGC     174   181   188   195   353   597   603   609   615   624   630   636   642   651
          657   663   669   678   684   690   696   705   711   717   723   732   738   744
          750   759   765   771   777   786   792   798   804   813   819   825   831   840
          846   852   858   867   873   879   885   894   900   906   912   921   927   933
```

```
EQ 1361        939     948     954     960     966     975     981     987     993    1002    1008    1014    1020    1115    SEQ 1374
              1377    1473    1576    1672    1775    1870    1973    2054    2063    2072    2081    2090    2099    2108
              2117    2126    2135    2144    2153    2162    2171    2180    2189    2200    2211    2222    2233    2244
              2255    2266    2277    2288    2299    2310    2321    2332    2343    2354    2365
.MNAND        1314    1318    1322    1326    1330    1334    1338    1342    1346    1350    1354    1358    1362    1366
              1370    1374    1497    1499    1521    1523    1545    1547    1569    1571    1696    1698    1720    1722
              1744    1746    1768    1770    1894    1896    1918    1920    1942    1944    1966    1968
.MOENA         174     181     188     195     353     597     603     609     615     624     630     636     642     651
               657     663     669     678     684     690     696     705     711     717     723     732     738     744
               750     759     765     771     777     786     792     798     804     813     819     825     831     840
               846     852     858     867     873     879     885     894     900     906     912     921     927     933
               939     948     954     960     966     975     981     987     993    1002    1008    1014    1020    1377
              1576    1775    1973
.MOR           174     177     179     181     184     186     188     191     193     195     297     299     301     303
               305     312     314     316     318     320     327     329     331     333     335     342     344     346
               348     350     353     597     599     601     603     605     607     609     611     613     615     624
               626     628     630     632     634     636     638     640     642     651     653     655     657     659
               661     663     665     667     669     678     680     682     684     686     688     690     692     694
               696     705     707     709     711     713     715     717     719     721     723     732     734     736
               738     740     742     744     746     748     750     759     761     763     765     767     769     771
               773     775     777     786     788     790     792     794     796     798     800     802     804     813
               815     817     819     821     823     825     827     829     831     840     842     844     846     848
               850     852     854     856     858     867     869     871     873     875     877     879     881     883
               885     894     896     898     900     902     904     906     908     910     912     921     923     925
               927     929     931     933     935     937     939     948     950     952     954     956     958     960
               962     964     966     975     977     979     981     983     985     987     989     991     993    1002
              1004    1006    1008    1010    1012    1014    1016    1018    1020    1115    1184    1188    1192    1196
              1200    1204    1208    1212    1216    1220    1224    1228    1232    1236    1240    1244    1377    1473
              1485    1487    1509    1511    1533    1535    1557    1559    1576    1672    1684    1686    1708    1710
              1732    1734    1756    1758    1772    1775    1870    1882    1884    1906    1908    1930    1932    1954
              1956    1970    1973    2056    2065    2074    2083    2092    2101    2110    2119    2128    2137    2146
              2155    2164    2173    2182    2191    2202    2213    2224    2235    2246    2257    2268    2279    2290
              2301    2312    2323    2334    2345    2356    2367
.MPLUS         172     290     595     622     649     676     703     730     757     784     811     838     865     892
               919     946     973    1000    1119    1123    1127    1131    1135    1139    1143    1147    1151    1155
              1159    1163    1167    1171    1175    1179    1479    1481    1503    1505    1527    1529    1551    1553
              1678    1680    1702    1704    1726    1728    1750    1752    1876    1878    1900    1902    1924    1926
              1948    1950
.MRPCT         305     320     335     350    1573    1772    1970
.MSOA          170     184     186     288     597     603     609     615     624     630     636     642     651     657
               663     669     678     684     690     696     705     711     717     723     732     738     744     750
               759     765     771     777     786     792     798     804     813     819     825     831     840     846
               852     858     867     873     879     885     894     900     906     912     921     927     933     939
               948     954     960     966     975     981     987     993    1002    1008    1014    1020    1117    1119
              1121    1123    1125    1127    1129    1131    1133    1135    1137    1139    1141    1143    1145    1147
              1149    1151    1153    1155    1157    1159    1161    1163    1165    1167    1169    1171    1173    1175
              1177    1179    1182    1184    1186    1188    1190    1192    1194    1196    1198    1200    1202    1204
              1206    1208    1210    1212    1214    1216    1218    1220    1222    1224    1226    1228    1230    1232
              1234    1236    1238    1240    1242    1244    1247    1249    1251    1253    1255    1257    1259    1261
              1263    1265    1267    1269    1271    1273    1275    1277    1279    1281    1283    1285    1287    1289
              1291    1293    1295    1297    1299    1301    1303    1305    1307    1309    1312    1314    1316    1318
              1320    1322    1324    1326    1328    1330    1332    1334    1336    1338    1340    1342    1344    1346
              1348    1350    1352    1354    1356    1358    1360    1362    1364    1366    1368    1370    1372    1374
```

```
          1377  1477  1479  1483  1485  1489  1491  1495  1497  1501  1503  1507  1509  1513
          1515  1519  1521  1525  1527  1531  1533  1537  1539  1543  1545  1549  1551  1555
          1557  1561  1563  1567  1569  1576  1676  1680  1682  1686  1688  1692  1694  1698
          1700  1704  1706  1710  1712  1716  1718  1722  1724  1728  1730  1734  1736  1740
          1742  1746  1748  1752  1754  1758  1760  1764  1766  1770  1775  1874  1876  1880
          1882  1886  1888  1892  1894  1898  1900  1904  1906  1910  1912  1916  1918  1922
          1924  1928  1930  1934  1936  1940  1942  1946  1948  1952  1954  1958  1960  1964
          1966  1973  2047  2050  2052  2056  2061  2065  2070  2074  2079  2083  2088  2092
          2097  2101  2106  2110  2115  2119  2124  2128  2133  2137  2142  2146  2151  2155
          2160  2164  2169  2173  2178  2182  2187  2191  2196  2198  2202  2207  2209  2213
          2218  2220  2224  2229  2231  2235  2240  2242  2246  2251  2253  2257  2262  2264
          2268  2273  2275  2279  2284  2286  2290  2295  2297  2301  2306  2308  2312  2317
          2319  2323  2328  2330  2334  2339  2341  2345  2350  2352  2356  2361  2363  2367
.MSOB      593   595   599   601   611   613   620   622   626   628   638   640   647   649
           653   655   665   667   674   676   680   682   692   694   701   703   707   709
           719   721   728   730   734   736   746   748   755   757   761   763   773   775
           782   784   788   790   800   802   809   811   815   817   827   829   836   838
           842   844   854   856   863   865   869   871   881   883   890   892   896   898
           908   910   917   919   923   925   935   937   944   946   950   952   962   964
           971   973   977   979   989   991   998  1000  1004  1006  1016  1018  1678  1684
          1690  1696  1702  1708  1714  1720  1726  1732  1738  1744  1750  1756  1762  1768
          1772  1970
.MSOQ      172   174   177   179   181   188   191   193   195   290   297   299   301   303
           305   312   314   316   318   320   327   329   331   333   335   342   344   346
           348   350   353   605   607   632   634   659   661   686   688   713   715   740
           742   767   769   794   796   821   823   848   850   875   877   902   904   929
           931   956   958   983   985  1010  1012  1481  1487  1493  1499  1505  1511  1517
          1523  1529  1535  1541  1547  1553  1559  1565  1571  1878  1884  1890  1896  1902
          1908  1914  1920  1926  1932  1938  1944  1950  1956  1962  1968
.MSDO     1115  1473  1672  1870
.MSDA     2054  2063  2072  2081  2090  2099  2108  2117  2126  2135  2144  2153  2162  2171
          2180  2189  2200  2211  2222  2233  2244  2255  2266  2277  2288  2299  2310  2321
          2332  2343  2354  2365
.MSELE     174   181   188   195   353   597   603   609   615   624   630   636   642   651
           657   663   669   678   684   690   696   705   711   717   723   732   738   744
           750   759   765   771   777   786   792   798   804   813   819   825   831   840
           846   852   858   867   873   879   885   894   900   906   912   921   927   933
           939   948   954   960   966   975   981   987   993  1002  1008  1014  1020  1377
          1576  1775  1973
.MSKCN    1115  1473  1672  1870  2054  2063  2072  2081  2090  2099  2108  2117  2126  2135
          2144  2153  2162  2171  2180  2189  2200  2211  2222  2233  2244  2255  2266  2277
          2288  2299  2310  2321  2332  2343  2354  2365
.MXNOR    2052  2061  2070  2079  2088  2097  2106  2115  2124  2133  2142  2151  2160  2169
          2178  2187  2198  2209  2220  2231  2242  2253  2264  2275  2286  2297  2308  2319
          2330  2341  2352  2363
.MXOR     1249  1253  1257  1261  1265  1269  1273  1277  1281  1285  1289  1293  1297  1301
          1305  1309  1491  1493  1515  1517  1539  1541  1563  1565  1690  1692  1714  1716
          1738  1740  1762  1764  1888  1890  1912  1914  1936  1938  1960  1962
.RA        184   186   597   603   609   615   624   630   636   642   651   657   663   669
           678   684   690   696   705   711   717   723   732   738   744   750   759   765
           771   777   786   792   798   804   813   819   825   831   840   845   852   858
           867   873   879   885   894   900   906   912   921   927   933   939   948   954
           960   966   975   981   987   993  1002  1008  1014  1020  1119  1123  1127  1131
```

```
SEQ 1363  1135  1139  1143  1147  1151  1155  1159  1163  1167  1171  1175  1179  1184  1188     SEQ 1376   D
          1192  1196  1200  1204  1208  1212  1216  1220  1224  1228  1232  1236  1240  1244                D
          1249  1253  1257  1261  1265  1269  1273  1277  1281  1285  1289  1293  1297  1301
          1305  1309  1314  1318  1322  1326  1330  1334  1338  1342  1346  1350  1354  1358
          1362  1366  1370  1374  1377  1479  1485  1491  1497  1503  1509  1515  1521  1527
          1533  1539  1545  1551  1557  1563  1569  1571  1576  1680  1770  1775  1876  1882
          1888  1894  1900  1906  1912  1918  1924  1930  1936  1942  1948  1954  1960  1966
          1968  1973  2047  2050  2052  2054  2056  2061  2063  2065  2070  2072  2074  2079
          2081  2083  2088  2090  2092  2097  2099  2101  2106  2108  2110  2115  2117  2119
          2124  2126  2128  2133  2135  2137  2142  2144  2146  2151  2153  2155  2160  2162
          2164  2169  2171  2173  2178  2180  2182  2187  2189  2191  2196  2198  2200  2202
          2207  2209  2211  2213  2218  2220  2222  2224  2229  2231  2233  2235  2240  2242
          2244  2246  2251  2253  2255  2257  2262  2264  2266  2268  2273  2275  2277  2279
          2284  2286  2288  2290  2295  2297  2299  2301  2306  2308  2310  2312  2317  2319
          2321  2323  2328  2330  2332  2334  2339  2341  2343  2345  2350  2352  2354  2356
          2361  2363  2365  2367
.RAND      170   288   593   620   647   674   701   728   755   782   809   836   863   890
           917   944   971   998  1117  1121  1125  1129  1133  1137  1141  1145  1149  1153
          1157  1161  1165  1169  1173  1177  1182  1186  1190  1194  1198  1202  1206  1210
          1214  1218  1222  1226  1230  1234  1238  1242  1247  1251  1255  1259  1263  1267
          1271  1275  1279  1283  1287  1291  1295  1299  1303  1307  1312  1316  1320  1324
          1328  1332  1336  1340  1344  1348  1352  1356  1360  1364  1368  1372  1477  1483
          1489  1495  1501  1507  1513  1519  1525  1531  1537  1543  1549  1555  1561  1567
          1676  1682  1688  1694  1700  1706  1712  1718  1724  1730  1736  1742  1748  1754
          1760  1766  1874  1880  1886  1892  1898  1904  1910  1916  1922  1928  1934  1940
          1946  1952  1958  1964  2047  2050  2054  2063  2072  2081  2090  2099  2108  2117
          2126  2135  2144  2153  2162  2171  2180  2189  2196  2200  2207  2211  2218  2222
          2229  2233  2240  2244  2251  2255  2262  2266  2273  2277  2284  2288  2295  2299
          2306  2310  2317  2321  2328  2332  2339  2343  2350  2354  2361  2365
.RB        177   191   593   595   599   601   605   607   611   613   620   622   626   628
           632   634   638   640   647   649   653   655   659   661   665   667   674   676
           680   682   686   688   692   694   701   703   707   709   713   715   719   721
           728   730   734   736   740   742   746   748   755   757   761   763   767   769
           773   775   782   784   788   790   794   796   800   802   809   811   815   817
           821   823   827   829   836   838   842   844   848   850   854   856   863   865
           869   871   875   877   881   883   890   892   896   898   902   904   908   910
           917   919   923   925   929   931   935   937   944   946   950   952   956   958
           962   964   971   973   977   979   983   985   989   991   998  1000  1004  1006
          1010  1012  1016  1018  1117  1119  1121  1123  1125  1127  1129  1131  1133  1135
          1137  1139  1141  1143  1145  1147  1149  1151  1153  1155  1157  1159  1161  1163
          1165  1167  1169  1171  1173  1175  1177  1179  1182  1184  1186  1188  1190  1192
          1194  1196  1198  1200  1202  1204  1206  1208  1210  1212  1214  1216  1218  1220
          1222  1224  1226  1228  1230  1232  1234  1236  1238  1240  1242  1244  1247  1249
          1251  1253  1255  1257  1259  1261  1263  1265  1267  1269  1271  1273  1275  1277
          1279  1281  1283  1285  1287  1289  1291  1293  1295  1297  1299  1301  1303  1305
          1307  1309  1312  1314  1316  1318  1320  1322  1324  1326  1328  1330  1332  1334
          1336  1338  1340  1342  1344  1346  1348  1350  1352  1354  1356  1358  1360  1362
          1364  1366  1368  1370  1372  1374  1477  1481  1483  1487  1489  1493  1495  1499
          1501  1505  1507  1511  1513  1517  1519  1523  1525  1529  1531  1535  1537  1541
          1543  1547  1549  1553  1555  1559  1561  1565  1567  1571  1676  1678  1680  1682
          1684  1686  1688  1690  1692  1694  1696  1698  1700  1702  1704  1706  1708  1710
          1712  1714  1716  1718  1720  1722  1724  1726  1728  1730  1732  1734  1736  1738
          1740  1742  1744  1746  1748  1750  1752  1754  1756  1758  1760  1762  1764  1766
```

```
SEQ 1364          1768  1770  1772  1874  1876  1878  1880  1882  1884  1886  1888  1890  1892  1894     SEQ 1377
                  1896  1898  1900  1902  1904  1906  1908  1910  1912  1914  1916  1918  1920  1922
                  1924  1926  1928  1930  1932  1934  1936  1938  1940  1942  1944  1946  1948  1950
                  1952  1954  1956  1958  1960  1962  1964  1966  1968  1970  2047  2050  2052  2054
                  2061  2063  2070  2072  2079  2081  2088  2090  2097  2099  2106  2108  2115  2117
                  2124  2126  2133  2135  2142  2144  2151  2153  2160  2162  2169  2171  2178  2180
                  2187  2189  2196  2198  2200  2207  2209  2211  2218  2220  2222  2229  2231  2233
                  2240  2242  2244  2251  2253  2255  2262  2264  2266  2273  2275  2277  2284  2286
                  2288  2295  2297  2299  2306  2308  2310  2317  2319  2321  2328  2330  2332  2339
                  2341  2343  2350  2352  2354  2361  2363  2365
        .RBAD     2374  2379
        .RCCFZ    2056  2065  2074  2083  2092  2101  2110  2119  2128  2137  2146  2155  2164  2173
                  2182  2191  2202  2213  2224  2235  2246  2257  2268  2279  2290  2301  2312  2323
                  2334  2345  2356  2367
        .RCENA    2056  2065  2074  2083  2092  2101  2110  2119  2128  2137  2146  2155  2164  2173
                  2182  2191  2202  2213  2224  2235  2246  2257  2268  2279  2290  2301  2312  2323
                  2334  2345  2356  2367
        .RCJP     2056  2065  2074  2083  2092  2101  2110  2119  2128  2137  2146  2155  2164  2173
                  2182  2191  2202  2213  2224  2235  2246  2257  2268  2279  2290  2301  2312  2323
                  2334  2345  2356  2367
        .RCONT     172   179   186   193   290   297   299   301   303   312   314   316   318   327
                   329   331   333   342   344   346   348   595   601   607   613   622   628   634
                   640   649   655   661   667   676   682   688   694   703   709   715   721   730
                   736   742   748   757   763   769   775   784   790   796   802   811   817   823
                   829   838   844   850   856   865   871   877   883   892   898   904   910   919
                   925   931   937   946   952   958   964   973   979   985   991  1000  1006  1012
                  1018  1117  1119  1121  1123  1125  1127  1129  1131  1133  1135  1137  1139  1141
                  1143  1145  1147  1149  1151  1153  1155  1157  1159  1161  1163  1165  1167  1169
                  1171  1173  1175  1177  1179  1182  1184  1186  1188  1190  1192  1194  1196  1198
                  1200  1202  1204  1206  1208  1210  1212  1214  1216  1218  1220  1222  1224  1226
                  1228  1230  1232  1234  1236  1238  1240  1242  1244  1247  1249  1251  1253  1255
                  1257  1259  1261  1263  1265  1267  1269  1271  1273  1275  1277  1279  1281  1283
                  1285  1287  1289  1291  1293  1295  1297  1299  1301  1303  1305  1307  1309  1312
                  1314  1316  1318  1320  1322  1324  1326  1328  1330  1332  1334  1336  1338  1340
                  1342  1344  1346  1348  1350  1352  1354  1356  1358  1360  1362  1364  1366  1368
                  1370  1372  1374  1477  1479  1481  1483  1485  1487  1489  1491  1493  1495  1497
                  1499  1501  1503  1505  1507  1509  1511  1513  1515  1517  1519  1521  1523  1525
                  1527  1529  1531  1533  1535  1537  1539  1541  1543  1545  1547  1549  1551  1553
                  1555  1557  1559  1561  1563  1565  1567  1569  1571  1676  1678  1680  1682  1684
                  1686  1688  1690  1692  1694  1696  1698  1700  1702  1704  1706  1708  1710  1712
                  1714  1716  1718  1720  1722  1724  1726  1728  1730  1732  1734  1736  1738  1740
                  1742  1744  1746  1748  1750  1752  1754  1756  1758  1760  1762  1764  1766  1768
                  1770  1874  1876  1878  1880  1882  1884  1886  1888  1890  1892  1894  1896  1898
                  1900  1902  1904  1906  1908  1910  1912  1914  1916  1918  1920  1922  1924  1926
                  1928  1930  1932  1934  1936  1938  1940  1942  1944  1946  1948  1950  1952  1954
                  1956  1958  1960  1962  1964  1966  1968  2052  2054  2061  2063  2070  2072  2079
                  2081  2088  2090  2097  2099  2106  2108  2115  2117  2124  2126  2133  2135  2142
                  2144  2151  2153  2160  2162  2169  2171  2178  2180  2187  2189  2196  2198  2200
                  2207  2209  2211  2218  2220  2222  2229  2231  2233  2240  2242  2244  2251  2253
                  2255  2262  2264  2266  2273  2275  2277  2284  2286  2288  2295  2297  2299  2306
                  2308  2310  2317  2319  2321  2328  2330  2332  2339  2341  2343  2350  2352  2354
                  2361  2363  2365
        .RCRY      172   290   595   622   649   676   703   730   757   784   811   838   865   892
```

```
                919   946   973  1000
SEQ 1365 .RD    170   172   174   177   179   181   184   186   188   191   193   195   288   290   SEQ 1378
                295   297   299   301   303   305   310   312   314   316   318   320   325   327
                329   331   333   335   340   342   344   346   348   350   353   593   595   597
                599   601   603   605   607   609   611   613   615   620   622   624   626   628
                630   632   634   636   638   640   642   647   649   651   653   655   657   659
                661   663   665   667   669   674   676   678   680   682   684   686   688   690
                692   694   696   701   703   705   707   709   711   713   715   717   719   721
                723   728   730   732   734   736   738   740   742   744   746   748   750   755
                757   759   761   763   765   767   769   771   773   775   777   782   784   786
                788   790   792   794   796   798   800   802   804   809   811   813   815   817
                819   821   823   825   827   829   831   836   838   840   842   844   846   848
                850   852   854   856   858   863   865   867   869   871   873   875   877   879
                881   883   885   890   892   894   896   898   900   902   904   906   908   910
                912   917   919   921   923   925   927   929   931   933   935   937   939   944
                946   948   950   952   954   956   958   960   962   964   966   971   973   975
                977   979   981   983   985   987   989   991   993   998  1000  1002  1004  1006
               1008  1010  1012  1014  1016  1018  1020  1115  1117  1119  1121  1123  1125  1127
               1129  1131  1133  1135  1137  1139  1141  1143  1145  1147  1149  1151  1153  1155
               1157  1159  1161  1163  1165  1167  1169  1171  1173  1175  1177  1179  1182  1184
               1186  1188  1190  1192  1194  1196  1198  1200  1202  1204  1206  1208  1210  1212
               1214  1216  1218  1220  1222  1224  1226  1228  1230  1232  1234  1236  1238  1240
               1242  1244  1247  1249  1251  1253  1255  1257  1259  1261  1263  1265  1267  1269
               1271  1273  1275  1277  1279  1281  1283  1285  1287  1289  1291  1293  1295  1297
               1299  1301  1303  1305  1307  1309  1312  1314  1316  1318  1320  1322  1324  1326
               1328  1330  1332  1334  1336  1338  1340  1342  1344  1346  1348  1350  1352  1354
               1356  1358  1360  1362  1364  1366  1368  1370  1372  1374  1473  1475  1477  1479
               1481  1483  1485  1487  1489  1491  1493  1495  1497  1499  1501  1503  1505  1507
               1509  1511  1513  1515  1517  1519  1521  1523  1525  1527  1529  1531  1533  1535
               1537  1539  1541  1543  1545  1547  1549  1551  1553  1555  1557  1559  1561  1563
               1565  1567  1569  1571  1573  1672  1674  1676  1678  1680  1682  1684  1686  1688
               1690  1692  1694  1696  1698  1700  1702  1704  1706  1708  1710  1712  1714  1716
               1718  1720  1722  1724  1726  1728  1730  1732  1734  1736  1738  1740  1742  1744
               1746  1748  1750  1752  1754  1756  1758  1760  1762  1764  1766  1768  1770  1772
               1870  1872  1874  1876  1878  1880  1882  1884  1886  1888  1890  1892  1894  1896
               1898  1900  1902  1904  1906  1908  1910  1912  1914  1916  1918  1920  1922  1924
               1926  1928  1930  1932  1934  1936  1938  1940  1942  1944  1946  1948  1950  1952
               1954  1956  1958  1960  1962  1964  1966  1968  1970  1973  2047  2050  2052  2054
               2061  2063  2070  2072  2079  2081  2088  2090  2097  2099  2106  2108  2115  2117
               2124  2126  2133  2135  2142  2144  2151  2153  2160  2162  2169  2171  2178  2180
               2187  2189  2196  2198  2200  2207  2209  2211  2218  2220  2222  2229  2231  2233
               2240  2242  2244  2251  2253  2255  2262  2264  2266  2273  2275  2277  2284  2286
               2288  2295  2297  2299  2306  2308  2310  2317  2319  2321  2328  2330  2332  2339
               2341  2343  2350  2352  2354  2361  2363  2365
         .RJ    170   174   177   181   184   188   191   195   288   295   305   310   320   325
                335   340   350   353   593   597   599   603   605   609   611   615   620   624
                626   630   632   636   638   642   647   651   653   657   659   663   665   669
                674   678   680   684   686   690   692   696   701   705   707   711   713   717
                719   723   728   732   734   738   740   744   746   750   755   759   761   765
                767   771   773   777   782   786   788   792   794   798   800   804   809   813
                815   819   821   825   827   831   836   840   842   846   848   852   854   858
                863   867   869   873   875   879   881   885   890   894   896   900   902   906
                908   912   917   921   923   927   929   933   935   939   944   948   950   954
```

```
            956   960   962   966   971   975   977   981   983   987   989   993   998  1002
           1004  1008  1010  1014  1016  1020  1115  1377  1379  1473  1475  1573  1576  1578     SEQ 1379
           1672  1674  1772  1775  1777  1870  1872  1970  1973  2045  2050  2056  2058  2065
           2067  2074  2076  2083  2085  2092  2094  2101  2103  2110  2112  2119  2121  2128
           2130  2137  2139  2146  2148  2155  2157  2164  2166  2173  2175  2182  2184  2191
           2193  2202  2204  2213  2215  2224  2226  2235  2237  2246  2248  2257  2259  2268
           2270  2279  2281  2290  2292  2301  2303  2312  2314  2323  2325  2334  2336  2345
           2347  2356  2358  2367  2369  2374  2379
 .RJMAP     170   174   177   181   184   188   191   195   288   353   593   597   599   603
            605   609   611   615   620   624   626   630   632   636   638   642   647   651
            653   657   659   663   665   669   674   678   680   684   686   690   692   696
            701   705   707   711   713   717   719   723   728   732   734   738   740   744
            746   750   755   759   761   765   767   771   773   777   782   786   788   792
            794   798   800   804   809   813   815   819   821   825   827   831   836   840
            842   846   848   852   854   858   863   867   869   873   875   879   881   885
            890   894   896   900   902   906   908   912   917   921   923   927   929   933
            935   939   944   948   950   954   956   960   962   966   971   975   977   981
            983   987   989   993   998  1002  1004  1008  1010  1014  1016  1020  1115  1377
           1379  1473  1576  1578  1672  1775  1777  1870  1973  2045  2050  2058  2067  2076
           2085  2094  2103  2112  2121  2130  2139  2148  2157  2166  2175  2184  2193  2204
           2215  2226  2237  2248  2259  2270  2281  2292  2303  2314  2325  2336  2347  2358
           2369  2374  2379
 .RJZ      2047
 .RLDCT     295   310   325   340  1475  1674  1872
 .RMGC      174   181   188   195   353   597   603   609   615   624   630   636   642   651
            657   663   669   678   684   690   696   705   711   717   723   732   738   744
            750   759   765   771   777   786   792   798   804   813   819   825   831   840
            846   852   858   867   873   879   885   894   900   906   912   921   927   933
            939   948   954   960   966   975   981   987   993  1002  1008  1014  1020  1115
           1377  1473  1576  1672  1775  1870  1973  2054  2063  2072  2081  2090  2099  2108
           2117  2126  2135  2144  2153  2162  2171  2180  2189  2200  2211  2222  2233  2244
           2255  2266  2277  2288  2299  2310  2321  2332  2343  2354  2365
 .RNAND    1314  1318  1322  1326  1330  1334  1338  1342  1346  1350  1354  1358  1362  1366
           1370  1374  1497  1499  1521  1523  1545  1547  1569  1571  1696  1698  1720  1722
           1744  1746  1768  1770  1894  1896  1918  1920  1942  1944  1966  1968
 .ROENA     174   181   188   195   353   597   603   609   615   624   630   636   642   651
            657   663   669   678   684   690   696   705   711   717   723   732   738   744
            750   759   765   771   777   786   792   798   804   813   819   825   831   840
            846   852   858   867   873   879   885   894   900   906   912   921   927   933
            939   948   954   960   966   975   981   987   993  1002  1008  1014  1020  1377
           1576  1775  1973
 .ROR       174   177   179   181   184   186   188   191   193   195   297   299   301   303
            305   312   314   316   318   320   327   329   331   333   335   342   344   346
            348   350   353   597   599   601   603   605   607   609   611   613   615   624
            626   628   630   632   634   636   638   640   642   651   653   655   657   659
            661   663   665   667   669   678   680   682   684   686   688   690   692   694
            696   705   707   709   711   713   715   717   719   721   723   732   734   736
            738   740   742   744   746   748   750   759   761   763   765   767   769   771
            773   775   777   786   788   790   792   794   796   798   800   802   804   813
            815   817   819   821   823   825   827   829   831   840   842   844   846   848
            850   852   854   856   858   867   869   871   873   875   877   879   881   883
            885   894   896   898   900   902   904   906   908   910   912   921   923   925
            927   929   931   933   935   937   939   948   950   952   954   956   958   960
```

```
            962   964   966   975   977   979   981   983   985   987   989   991   993  1002
```

1367

```
             962   964   966   975   977   979   981   983   985   987   989   991   993  1002      SEQ 1380
            1004  1006  1008  1010  1012  1014  1016  1018  1020  1115  1184  1188  1192  1196
            1200  1204  1208  1212  1216  1220  1224  1228  1232  1236  1240  1244  1377  1473
            1485  1487  1509  1511  1533  1535  1557  1559  1576  1672  1684  1686  1708  1710
            1732  1734  1756  1758  1772  1775  1870  1882  1884  1906  1908  1930  1932  1954
            1956  1970  1973  2056  2065  2074  2083  2092  2101  2110  2119  2128  2137  2146
            2155  2164  2173  2182  2191  2202  2213  2224  2235  2246  2257  2268  2279  2290
            2301  2312  2323  2334  2345  2356  2367
.RPLUS       172   290   595   622   649   676   703   730   757   784   811   838   865   892
             919   946   973  1000  1119  1123  1127  1131  1135  1139  1143  1147  1151  1155
            1159  1163  1167  1171  1175  1179  1479  1481  1503  1505  1527  1529  1551  1553
            1678  1680  1702  1704  1726  1728  1750  1752  1876  1878  1900  1902  1924  1926
            1948  1950
.RRPCT       305   320   335   350  1573  1772  1970
.RSOA        170   184   186   288   597   603   609   615   624   630   636   642   651   657
             663   669   678   684   690   696   705   711   717   723   732   738   744   750
             759   765   771   777   786   792   798   804   813   819   825   831   840   846
             852   858   867   873   879   885   894   900   906   912   921   927   933   939
             948   954   960   966   975   981   987   993  1002  1008  1014  1020  1117  1119
            1121  1123  1125  1127  1129  1131  1133  1135  1137  1139  1141  1143  1145  1147
            1149  1151  1153  1155  1157  1159  1161  1163  1165  1167  1169  1171  1173  1175
            1177  1179  1182  1184  1186  1188  1190  1192  1194  1196  1198  1200  1202  1204
            1206  1208  1210  1212  1214  1216  1218  1220  1222  1224  1226  1228  1230  1232
            1234  1236  1238  1240  1242  1244  1247  1249  1251  1253  1255  1257  1259  1261
            1263  1265  1267  1269  1271  1273  1275  1277  1279  1281  1283  1285  1287  1289
            1291  1293  1295  1297  1299  1301  1303  1305  1307  1309  1312  1314  1316  1318
            1320  1322  1324  1326  1328  1330  1332  1334  1336  1338  1340  1342  1344  1346
            1348  1350  1352  1354  1356  1358  1360  1362  1364  1366  1368  1370  1372  1374
            1377  1477  1479  1483  1485  1489  1491  1495  1497  1501  1503  1507  1509  1513
            1515  1519  1521  1525  1527  1531  1533  1537  1539  1543  1545  1549  1551  1555
            1557  1561  1563  1567  1569  1576  1676  1680  1682  1686  1688  1692  1694  1698
            1700  1704  1706  1710  1712  1716  1718  1722  1724  1728  1730  1734  1736  1740
            1742  1746  1748  1752  1754  1758  1760  1764  1766  1770  1775  1874  1876  1880
            1882  1886  1888  1892  1894  1898  1900  1904  1906  1910  1912  1916  1918  1922
            1924  1928  1930  1934  1936  1940  1942  1946  1948  1952  1954  1958  1960  1964
            1966  1973  2047  2050  2052  2056  2061  2065  2070  2074  2079  2083  2088  2092
            2097  2101  2106  2110  2115  2119  2124  2128  2133  2137  2142  2146  2151  2155
            2160  2164  2169  2173  2178  2182  2187  2191  2196  2198  2202  2207  2209  2213
            2218  2220  2224  2229  2231  2235  2240  2242  2246  2251  2253  2257  2262  2264
            2268  2273  2275  2279  2284  2286  2290  2295  2297  2301  2306  2308  2312  2317
            2319  2323  2328  2330  2334  2339  2341  2345  2350  2352  2356  2361  2363  2367
.RSOB        593   595   599   601   611   613   620   622   626   628   638   640   647   649
             653   655   665   667   674   676   680   682   692   694   701   703   707   709
             719   721   728   730   734   736   746   748   755   757   761   763   773   775
             782   784   788   790   800   802   809   811   815   817   827   829   836   838
             842   844   854   856   863   865   869   871   881   883   890   892   896   898
             908   910   917   919   923   925   935   937   944   946   950   952   962   964
             971   973   977   979   989   991   998  1000  1004  1006  1016  1018  1678  1684
            1690  1696  1702  1708  1714  1720  1726  1732  1738  1744  1750  1756  1762  1768
            1772  1970
.RSOG        172   174   177   179   181   188   191   193   195   290   297   299   301   303
             305   312   314   316   318   320   327   329   331   333   335   342   344   346
             348   350   353   605   607   632   634   659   661   686   688   713   715   740
```

DFPTA
DFPTA

1368

```
             742   767   769   794   796   821   823   848   850   875   877   902   904   929      SEQ 1381
```

DFPTA

```
1368            742   767   769   794   796   821   823   848   850   875   877   902   904   929      SEQ 1381
                931   956   958   983   985  1010  1012  1481  1487  1493  1499  1505  1511  1517
               1523  1529  1535  1541  1547  1553  1559  1565  1571  1878  1884  1890  1896  1902
               1908  1914  1920  1926  1932  1938  1944  1950  1956  1962  1968
     .RSDO     1115  1473  1672  1870
     .RSDA     2054  2063  2072  2081  2090  2099  2108  2117  2126  2135  2144  2153  2162  2171
               2180  2189  2200  2211  2222  2233  2244  2255  2266  2277  2288  2299  2310  2321
               2332  2343  2354  2365
     .RSELE     174   181   188   195   353   597   603   609   615   624   630   636   642   651
                657   663   669   678   684   690   696   705   711   717   723   732   738   744
                750   759   765   771   777   786   792   798   804   813   819   825   831   840
                846   852   858   867   873   879   885   894   900   906   912   921   927   933
                939   948   954   960   966   975   981   987   993  1002  1008  1014  1020  1377
               1576  1775  1973
     .RSKCN    1115  1473  1672  1870  2054  2063  2072  2081  2090  2099  2108  2117  2126  2135
               2144  2153  2162  2171  2180  2189  2200  2211  2222  2233  2244  2255  2266  2277
               2288  2299  2310  2321  2332  2343  2354  2365
     .RXNOR    2052  2061  2070  2079  2088  2097  2106  2115  2124  2133  2142  2151  2160  2169
               2178  2187  2198  2209  2220  2231  2242  2253  2264  2275  2286  2297  2308  2319
               2330  2341  2352  2363
     .RXOR     1249  1253  1257  1261  1265  1269  1273  1277  1281  1285  1289  1293  1297  1301
               1305  1309  1491  1493  1515  1517  1539  1541  1563  1565  1690  1692  1714  1716
               1738  1740  1762  1764  1888  1890  1912  1914  1936  1938  1960  1962
```

```
1369 ATABLE     119   121   122   123   125   126   445   446   447   449   450   195   288   290      SEQ 1382
```

```
1369  ATABLE     119   121   122   123   125   126   445   446   447   449   450                      SEQ 1382     DFPTA
      CALC       170   172   174   177   179   181   184   186   188   191   193   195   288   290                 DFPTA
                 295   297   299   301   303   305   310   312   314   316   318   320   325   327
                 329   331   333   335   340   342   344   346   348   350   353   593   595   597
                 599   601   603   605   607   609   611   613   615   620   622   624   626   628
                 630   632   634   636   638   640   642   647   649   651   653   655   657   659
                 661   663   665   667   669   674   676   678   680   682   684   686   688   690
                 692   694   696   701   703   705   707   709   711   713   715   717   719   721
                 723   728   730   732   734   736   738   740   742   744   746   748   750   755
                 757   759   761   763   765   767   769   771   773   775   777   782   784   786
                 788   790   792   794   796   798   800   802   804   809   811   813   815   817
                 819   821   823   825   827   829   831   836   838   840   842   844   846   848
                 850   852   854   856   858   863   865   867   869   871   873   875   877   879
                 881   883   885   890   892   894   896   898   900   902   904   906   908   910
                 912   917   919   921   923   925   927   929   931   933   935   937   939   944
                 946   948   950   952   954   956   958   960   962   964   966   971   973   975
                 977   979   981   983   985   987   989   991   993   998  1000  1002  1004  1006
                1008  1010  1012  1014  1016  1018  1020  1115  1117  1119  1121  1123  1125  1127
                1129  1131  1133  1135  1137  1139  1141  1143  1145  1147  1149  1151  1153  1155
                1157  1159  1161  1163  1165  1167  1169  1171  1173  1175  1177  1179  1182  1184
                1186  1188  1190  1192  1194  1196  1198  1200  1202  1204  1206  1208  1210  1212
                1214  1216  1218  1220  1222  1224  1226  1228  1230  1232  1234  1236  1238  1240
                1242  1244  1247  1249  1251  1253  1255  1257  1259  1261  1263  1265  1267  1269
                1271  1273  1275  1277  1279  1281  1283  1285  1287  1289  1291  1293  1295  1297
                1299  1301  1303  1305  1307  1309  1312  1314  1316  1318  1320  1322  1324  1326
                1328  1330  1332  1334  1336  1338  1340  1342  1344  1346  1348  1350  1352  1354
                1356  1358  1360  1362  1364  1366  1368  1370  1372  1374  1377  1379  1473  1475
                1477  1479  1481  1483  1485  1487  1489  1491  1493  1495  1497  1499  1501  1503
                1505  1507  1509  1511  1513  1515  1517  1519  1521  1523  1525  1527  1529  1531
                1533  1535  1537  1539  1541  1543  1545  1547  1549  1551  1553  1555  1557  1559
                1561  1563  1565  1567  1569  1571  1573  1576  1578  1672  1674  1676  1678  1680
                1682  1684  1686  1688  1690  1692  1694  1696  1698  1700  1702  1704  1706  1708
                1710  1712  1714  1716  1718  1720  1722  1724  1726  1728  1730  1732  1734  1736
                1738  1740  1742  1744  1746  1748  1750  1752  1754  1756  1758  1760  1762  1764
                1766  1768  1770  1772  1775  1777  1870  1872  1874  1876  1878  1880  1882  1884
                1886  1888  1890  1892  1894  1896  1898  1900  1902  1904  1906  1908  1910  1912
                1914  1916  1918  1920  1922  1924  1926  1928  1930  1932  1934  1936  1938  1940
                1942  1944  1946  1948  1950  1952  1954  1956  1958  1960  1962  1964  1966  1968
                1970  1973  2045  2047  2050  2052  2054  2056  2058  2061  2063  2065  2067  2070
                2072  2074  2076  2079  2081  2083  2085  2088  2090  2092  2094  2097  2099  2101
                2103  2106  2108  2110  2112  2115  2117  2119  2121  2124  2126  2128  2130  2133
                2135  2137  2139  2142  2144  2146  2148  2151  2153  2155  2157  2160  2162  2164
                2166  2169  2171  2173  2175  2178  2180  2182  2184  2187  2189  2191  2193  2196
                2198  2200  2202  2204  2207  2209  2211  2213  2215  2218  2220  2222  2224  2226
                2229  2231  2233  2235  2237  2240  2242  2244  2246  2248  2251  2253  2255  2257
                2259  2262  2264  2266  2268  2270  2273  2275  2277  2279  2281  2284  2286  2288
                2290  2292  2295  2297  2299  2301  2303  2306  2308  2310  2312  2314  2317  2319
                2321  2323  2325  2328  2330  2332  2334  2336  2339  2341  2343  2345  2347  2350
                2352  2354  2356  2358  2361  2363  2365  2367  2369  2374  2379
      CONCAT     170   172   174   177   179   181   184   186   188   191   193   195   288   290
                 295   297   299   301   303   305   310   312   314   316   318   320   325   327
                 329   331   333   335   340   342   344   346   348   350   353   593   595   597
                 599   601   603   605   607   609   611   613   615   620   622   624   626   628
```

```
1370             630   632   634   636   638   640   642   647   649   651   653   655   657   659     SEQ 1383     DFPT
```

```
1370      630   632   634   636   638   640   642   647   649   651   653   655   657   659    SEQ 1383    DFPT
          661   663   665   667   669   674   676   678   680   682   684   686   688   690                DFPT
          692   694   696   701   703   705   707   709   711   713   715   717   719   721
          723   728   730   732   734   736   738   740   742   744   746   748   750   755
          757   759   761   763   765   767   769   771   773   775   777   782   784   786
          788   790   792   794   796   798   800   802   804   809   811   813   815   817
          819   821   823   825   827   829   831   836   838   840   842   844   846   848
          850   852   854   856   858   863   865   867   869   871   873   875   877   879
          881   883   885   890   892   894   896   898   900   902   904   906   908   910
          912   917   919   921   923   925   927   929   931   933   935   937   939   944
          946   948   950   952   954   956   958   960   962   964   966   971   973   975
          977   979   981   983   985   987   989   991   993   998  1000  1002  1004  1006
         1008  1010  1012  1014  1016  1018  1020  1115  1117  1119  1121  1123  1125  1127
         1129  1131  1133  1135  1137  1139  1141  1143  1145  1147  1149  1151  1153  1155
         1157  1159  1161  1163  1165  1167  1169  1171  1173  1175  1177  1179  1182  1184
         1186  1188  1190  1192  1194  1196  1198  1200  1202  1204  1206  1208  1210  1212
         1214  1216  1218  1220  1222  1224  1226  1228  1230  1232  1234  1236  1238  1240
         1242  1244  1247  1249  1251  1253  1255  1257  1259  1261  1263  1265  1267  1269
         1271  1273  1275  1277  1279  1281  1283  1285  1287  1289  1291  1293  1295  1297
         1299  1301  1303  1305  1307  1309  1312  1314  1316  1318  1320  1322  1324  1326
         1328  1330  1332  1334  1336  1338  1340  1342  1344  1346  1348  1350  1352  1354
         1356  1358  1360  1362  1364  1366  1368  1370  1372  1374  1377  1379  1473  1475
         1477  1479  1481  1483  1485  1487  1489  1491  1493  1495  1497  1499  1501  1503
         1505  1507  1509  1511  1513  1515  1517  1519  1521  1523  1525  1527  1529  1531
         1533  1535  1537  1539  1541  1543  1545  1547  1549  1551  1553  1555  1557  1559
         1561  1563  1565  1567  1569  1571  1573  1576  1578  1672  1674  1676  1678  1680
         1682  1684  1686  1688  1690  1692  1694  1696  1698  1700  1702  1704  1706  1708
         1710  1712  1714  1716  1718  1720  1722  1724  1726  1728  1730  1732  1734  1736
         1738  1740  1742  1744  1746  1748  1750  1752  1754  1756  1758  1760  1762  1764
         1766  1768  1770  1772  1775  1777  1870  1872  1874  1876  1878  1880  1882  1884
         1886  1888  1890  1892  1894  1896  1898  1900  1902  1904  1906  1908  1910  1912
         1914  1916  1918  1920  1922  1924  1926  1928  1930  1932  1934  1936  1938  1940
         1942  1944  1946  1948  1950  1952  1954  1956  1958  1960  1962  1964  1966  1968
         1970  1973  2045  2047  2050  2052  2054  2056  2058  2061  2063  2065  2067  2070
         2072  2074  2076  2079  2081  2083  2085  2088  2090  2092  2094  2097  2099  2101
         2103  2106  2108  2110  2112  2115  2117  2119  2121  2124  2126  2128  2130  2133
         2135  2137  2139  2142  2144  2146  2148  2151  2153  2155  2157  2160  2162  2164
         2166  2169  2171  2173  2175  2178  2180  2182  2184  2187  2189  2191  2193  2196
         2198  2200  2202  2204  2207  2209  2211  2213  2215  2218  2220  2222  2224  2226
         2229  2231  2233  2235  2237  2240  2242  2244  2246  2248  2251  2253  2255  2257
         2259  2262  2264  2266  2268  2270  2273  2275  2277  2279  2281  2284  2286  2288
         2290  2292  2295  2297  2299  2301  2303  2306  2308  2310  2312  2314  2317  2319
         2321  2323  2325  2328  2330  2332  2334  2336  2339  2341  2343  2345  2347  2350
         2352  2354  2356  2358  2361  2363  2365  2367  2369  2374  2379
FIELD     170   172   174   177   179   181   184   186   188   191   193   195   288   290
          295   297   299   301   303   305   310   312   314   316   318   320   325   327
          329   331   333   335   340   342   344   346   348   350   353   593   595   597
          599   601   603   605   607   609   611   613   615   620   622   624   626   628
          630   632   634   636   638   640   642   647   649   651   653   655   657   659
          661   663   665   667   669   674   676   678   680   682   684   686   688   690
          692   694   696   701   703   705   707   709   711   713   715   717   719   721
          723   728   730   732   734   736   738   740   742   744   746   748   750   755
          757   759   761   763   765   767   769   771   773   775   777   782   784   786



          788   790   792   794   796   798   800   802   804   809   811   813   815   817                DFPT
          788    790    792    794    796    798    800    802    804    809    811    813    815    817     SEQ 1384
          819    821    823    825    827    829    831    836    838    840    842    844    846    848
          850    852    854    856    858    863    865    867    869    871    873    875    877    879
          881    883    885    890    892    894    896    898    900    902    904    906    908    910
          912    917    919    921    923    925    927    929    931    933    935    937    939    944
          946    948    950    952    954    956    958    960    962    964    966    971    973    975
          977    979    981    983    985    987    989    991    993    998   1000   1002   1004   1006
         1008   1010   1012   1014   1016   1018   1020   1115   1117   1119   1121   1123   1125   1127
         1129   1131   1133   1135   1137   1139   1141   1143   1145   1147   1149   1151   1153   1155
         1157   1159   1161   1163   1165   1167   1169   1171   1173   1175   1177   1179   1182   1184
         1186   1188   1190   1192   1194   1196   1198   1200   1202   1204   1206   1208   1210   1212
         1214   1216   1218   1220   1222   1224   1226   1228   1230   1232   1234   1236   1238   1240
         1242   1244   1247   1249   1251   1253   1255   1257   1259   1261   1263   1265   1267   1269
         1271   1273   1275   1277   1279   1281   1283   1285   1287   1289   1291   1293   1295   1297
         1299   1301   1303   1305   1307   1309   1312   1314   1316   1318   1320   1322   1324   1326
         1328   1330   1332   1334   1336   1338   1340   1342   1344   1346   1348   1350   1352   1354
         1356   1358   1360   1362   1364   1366   1368   1370   1372   1374   1377   1379   1473   1475
         1477   1479   1481   1483   1485   1487   1489   1491   1493   1495   1497   1499   1501   1503
         1505   1507   1509   1511   1513   1515   1517   1519   1521   1523   1525   1527   1529   1531
         1533   1535   1537   1539   1541   1543   1545   1547   1549   1551   1553   1555   1557   1559
         1561   1563   1565   1567   1569   1571   1573   1576   1578   1672   1674   1676   1678   1680
         1682   1684   1686   1688   1690   1692   1694   1696   1698   1700   1702   1704   1706   1708
         1710   1712   1714   1716   1718   1720   1722   1724   1726   1728   1730   1732   1734   1736
         1738   1740   1742   1744   1746   1748   1750   1752   1754   1756   1758   1760   1762   1764
         1766   1768   1770   1772   1775   1777   1870   1872   1874   1876   1878   1880   1882   1884
         1886   1888   1890   1892   1894   1896   1898   1900   1902   1904   1906   1908   1910   1912
         1914   1916   1918   1920   1922   1924   1926   1928   1930   1932   1934   1936   1938   1940
         1942   1944   1946   1948   1950   1952   1954   1956   1958   1960   1962   1964   1966   1968
         1970   1973   2045   2047   2050   2052   2054   2056   2058   2061   2063   2065   2067   2070
         2072   2074   2076   2079   2081   2083   2085   2088   2090   2092   2094   2097   2099   2101
         2103   2106   2108   2110   2112   2115   2117   2119   2121   2124   2126   2128   2130   2133
         2135   2137   2139   2142   2144   2146   2148   2151   2153   2155   2157   2160   2162   2164
         2166   2169   2171   2173   2175   2178   2180   2182   2184   2187   2189   2191   2193   2196
         2198   2200   2202   2204   2207   2209   2211   2213   2215   2218   2220   2222   2224   2226
         2229   2231   2233   2235   2237   2240   2242   2244   2246   2248   2251   2253   2255   2257
         2259   2262   2264   2266   2268   2270   2273   2275   2277   2279   2281   2284   2286   2288
         2290   2292   2295   2297   2299   2301   2303   2306   2308   2310   2312   2314   2317   2319
         2321   2323   2325   2328   2330   2332   2334   2336   2339   2341   2343   2345   2347   2350
         2352   2354   2356   2358   2361   2363   2365   2367   2369   2374   2379
GO         87     89     95    101    235    237    243    251    413    415    421    427   1059   1061
         1067   1075   1417   1419   1425   1433   1618   1620   1626   1634   1816   1818   1824   1832
         2009   2011   2017   2025
MFLD     170#    170   172#    172   174#    174   177#    177   179#    179   181#    181   184#    184
         186#    186   188#    188   191#    191   193#    193   195#    195   288#    288   290#    290
         295#    295   297#    297   299#    299   301#    301   303#    303   305#    305   310#    310
         312#    312   314#    314   316#    316   318#    318   320#    320   325#    325   327#    327
         329#    329   331#    331   333#    333   335#    335   340#    340   342#    342   344#    344
         346#    346   348#    348   350#    350   353#    353   593#    593   595#    595   597#    597
         599#    599   601#    601   603#    603   605#    605   607#    607   609#    609   611#    611
         613#    613   615#    615   620#    620   622#    622   624#    624   626#    626   628#    628
         630#    630   632#    632   634#    634   636#    636   638#    638   640#    640   642#    642
         647#    647   649#    649   651#    651   653#    653   655#    655   657#    657   659#    659
         661#    661   663#    663   665#    665   667#    667   669#    669   674#    674   676#    676
```

```
 678#   678    680#   680    682#   682    684#   684    686#   686    688#   688    690#   690     SEQ 1385
 692#   692    694#   694    696#   696    701#   701    703#   703    705#   705    707#   707
 709#   709    711#   711    713#   713    715#   715    717#   717    719#   719    721#   721
 723#   723    728#   728    730#   730    732#   732    734#   734    736#   736    738#   738
 740#   740    742#   742    744#   744    746#   746    748#   748    750#   750    755#   755
 757#   757    759#   759    761#   761    763#   763    765#   765    767#   767    769#   769
 771#   771    773#   773    775#   775    777#   777    782#   782    784#   784    786#   786
 788#   788    790#   790    792#   792    794#   794    796#   796    798#   798    800#   800
 802#   802    804#   804    809#   809    811#   811    813#   813    815#   815    817#   817
 819#   819    821#   821    823#   823    825#   825    827#   827    829#   829    831#   831
 836#   836    838#   838    840#   840    842#   842    844#   844    846#   846    848#   848
 850#   850    852#   852    854#   854    856#   856    858#   858    863#   863    865#   865
 867#   867    869#   869    871#   871    873#   873    875#   875    877#   877    879#   879
 881#   881    883#   883    885#   885    890#   890    892#   892    894#   894    896#   896
 898#   898    900#   900    902#   902    904#   904    906#   906    908#   908    910#   910
 912#   912    917#   917    919#   919    921#   921    923#   923    925#   925    927#   927
 929#   929    931#   931    933#   933    935#   935    937#   937    939#   939    944#   944
 946#   946    948#   948    950#   950    952#   952    954#   954    956#   956    958#   958
 960#   960    962#   962    964#   964    966#   966    971#   971    973#   973    975#   975
 977#   977    979#   979    981#   981    983#   983    985#   985    987#   987    989#   989
 991#   991    993#   993    998#   998   1000#  1000   1002#  1002   1004#  1004   1006#  1006
1008#  1008   1010#  1010   1012#  1012   1014#  1014   1016#  1016   1018#  1018   1020#  1020
1115#  1115   1117#  1117   1119#  1119   1121#  1121   1123#  1123   1125#  1125   1127#  1127
1129#  1129   1131#  1131   1133#  1133   1135#  1135   1137#  1137   1139#  1139   1141#  1141
1143#  1143   1145#  1145   1147#  1147   1149#  1149   1151#  1151   1153#  1153   1155#  1155
1157#  1157   1159#  1159   1161#  1161   1163#  1163   1165#  1165   1167#  1167   1169#  1169
1171#  1171   1173#  1173   1175#  1175   1177#  1177   1179#  1179   1182#  1182   1184#  1184
1186#  1186   1188#  1188   1190#  1190   1192#  1192   1194#  1194   1196#  1196   1198#  1198
1200#  1200   1202#  1202   1204#  1204   1206#  1206   1208#  1208   1210#  1210   1212#  1212
1214#  1214   1216#  1216   1218#  1218   1220#  1220   1222#  1222   1224#  1224   1226#  1226
1228#  1228   1230#  1230   1232#  1232   1234#  1234   1236#  1236   1238#  1238   1240#  1240
1242#  1242   1244#  1244   1247#  1247   1249#  1249   1251#  1251   1253#  1253   1255#  1255
1257#  1257   1259#  1259   1261#  1261   1263#  1263   1265#  1265   1267#  1267   1269#  1269
1271#  1271   1273#  1273   1275#  1275   1277#  1277   1279#  1279   1281#  1281   1283#  1283
1285#  1285   1287#  1287   1289#  1289   1291#  1291   1293#  1293   1295#  1295   1297#  1297
1299#  1299   1301#  1301   1303#  1303   1305#  1305   1307#  1307   1309#  1309   1312#  1312
1314#  1314   1316#  1316   1318#  1318   1320#  1320   1322#  1322   1324#  1324   1326#  1326
1328#  1328   1330#  1330   1332#  1332   1334#  1334   1336#  1336   1338#  1338   1340#  1340
1342#  1342   1344#  1344   1346#  1346   1348#  1348   1350#  1350   1352#  1352   1354#  1354
1356#  1356   1358#  1358   1360#  1360   1362#  1362   1364#  1364   1366#  1366   1368#  1368
1370#  1370   1372#  1372   1374#  1374   1377#  1377   1379#  1379   1473#  1473   1475#  1475
1477#  1477   1479#  1479   1481#  1481   1483#  1483   1485#  1485   1487#  1487   1489#  1489
1491#  1491   1493#  1493   1495#  1495   1497#  1497   1499#  1499   1501#  1501   1503#  1503
1505#  1505   1507#  1507   1509#  1509   1511#  1511   1513#  1513   1515#  1515   1517#  1517
1519#  1519   1521#  1521   1523#  1523   1525#  1525   1527#  1527   1529#  1529   1531#  1531
1533#  1533   1535#  1535   1537#  1537   1539#  1539   1541#  1541   1543#  1543   1545#  1545
1547#  1547   1549#  1549   1551#  1551   1553#  1553   1555#  1555   1557#  1557   1559#  1559
1561#  1561   1563#  1563   1565#  1565   1567#  1567   1569#  1569   1571#  1571   1573#  1573
1576#  1576   1578#  1578   1672#  1672   1674#  1674   1676#  1676   1678#  1678   1680#  1680
1682#  1682   1684#  1684   1686#  1686   1688#  1688   1690#  1690   1692#  1692   1694#  1694
1696#  1696   1698#  1698   1700#  1700   1702#  1702   1704#  1704   1706#  1705   1708#  1708
1710#  1710   1712#  1712   1714#  1714   1716#  1716   1718#  1718   1720#  1720   1722#  1722
1724#  1724   1726#  1726   1728#  1728   1730#  1730   1732#  1732   1734#  1734   1736#  1736
```

```
EQ 1373  1738#  1738   1740#  1740   1742#  1742   1744#  1744   1746#  1746   1748#  1748   1750#  1750     SEQ 1386   DFF
         1752#  1752   1754#  1754   1756#  1756   1758#  1758   1760#  1760   1762#  1762   1764#  1764                DFF
         1766#  1766   1768#  1768   1770#  1770   1772#  1772   1775#  1775   1777#  1777   1870#  1870
         1872#  1872   1874#  1874   1876#  1876   1878#  1878   1880#  1880   1882#  1882   1884#  1884
         1886#  1886   1888#  1888   1890#  1890   1892#  1892   1894#  1894   1896#  1896   1898#  1898
         1900#  1900   1902#  1902   1904#  1904   1906#  1906   1908#  1908   1910#  1910   1912#  1912
         1914#  1914   1916#  1916   1918#  1918   1920#  1920   1922#  1922   1924#  1924   1926#  1926
         1928#  1928   1930#  1930   1932#  1932   1934#  1934   1936#  1936   1938#  1938   1940#  1940
         1942#  1942   1944#  1944   1946#  1946   1948#  1948   1950#  1950   1952#  1952   1954#  1954
         1956#  1956   1958#  1958   1960#  1960   1962#  1962   1964#  1964   1966#  1966   1968#  1968
         1970#  1970   1973#  1973   2045#  2045   2047#  2047   2050#  2050   2052#  2052   2054#  2054
         2056#  2056   2058#  2058   2061#  2061   2063#  2063   2065#  2065   2067#  2067   2070#  2070
         2072#  2072   2074#  2074   2076#  2076   2079#  2079   2081#  2081   2083#  2083   2085#  2085
         2088#  2088   2090#  2090   2092#  2092   2094#  2094   2097#  2097   2099#  2099   2101#  2101
         2103#  2103   2106#  2106   2108#  2108   2110#  2110   2112#  2112   2115#  2115   2117#  2117
         2119#  2119   2121#  2121   2124#  2124   2126#  2126   2128#  2128   2130#  2130   2133#  2133
         2135#  2135   2137#  2137   2139#  2139   2142#  2142   2144#  2144   2146#  2146   2148#  2148
         2151#  2151   2153#  2153   2155#  2155   2157#  2157   2160#  2160   2162#  2162   2164#  2164
         2166#  2166   2169#  2169   2171#  2171   2173#  2173   2175#  2175   2178#  2178   2180#  2180
         2182#  2182   2184#  2184   2187#  2187   2189#  2189   2191#  2191   2193#  2193   2196#  2196
         2198#  2198   2200#  2200   2202#  2202   2204#  2204   2207#  2207   2209#  2209   2211#  2211
         2213#  2213   2215#  2215   2218#  2218   2220#  2220   2222#  2222   2224#  2224   2226#  2226
         2229#  2229   2231#  2231   2233#  2233   2235#  2235   2237#  2237   2240#  2240   2242#  2242
         2244#  2244   2246#  2246   2248#  2248   2251#  2251   2253#  2253   2255#  2255   2257#  2257
         2259#  2259   2262#  2262   2264#  2264   2266#  2266   2268#  2268   2270#  2270   2273#  2273
         2275#  2275   2277#  2277   2279#  2279   2281#  2281   2284#  2284   2286#  2286   2288#  2288
         2290#  2290   2292#  2292   2295#  2295   2297#  2297   2299#  2299   2301#  2301   2303#  2303
         2306#  2306   2308#  2308   2310#  2310   2312#  2312   2314#  2314   2317#  2317   2319#  2319
         2321#  2321   2323#  2323   2325#  2325   2328#  2328   2330#  2330   2332#  2332   2334#  2334
         2336#  2336   2339#  2339   2341#  2341   2343#  2343   2345#  2345   2347#  2347   2350#  2350
         2352#  2352   2354#  2354   2356#  2356   2358#  2358   2361#  2361   2363#  2363   2365#  2365
         2367#  2367   2369#  2369   2374#  2374   2379#  2379
MWORD     170    172    174    177    179    181    184    186    188    191    193    195    288    290
          295    297    299    301    303    305    310    312    314    316    318    320    325    327
          329    331    333    335    340    342    344    346    348    350    353    593    595    597
          599    601    603    605    607    609    611    613    615    620    622    624    626    628
          630    632    634    636    638    640    642    647    649    651    653    655    657    659
          661    663    665    667    669    674    676    678    680    682    684    686    688    690
          692    694    696    701    703    705    707    709    711    713    715    717    719    721
          723    728    730    732    734    736    738    740    742    744    746    748    750    755
          757    759    761    763    765    767    769    771    773    775    777    782    784    786
          788    790    792    794    796    798    800    802    804    809    811    813    815    817
          819    821    823    825    827    829    831    836    838    840    842    844    846    848
          850    852    854    856    858    863    865    867    869    871    873    875    877    879
          881    883    885    890    892    894    896    898    900    902    904    906    908    910
          912    917    919    921    923    925    927    929    931    933    935    937    939    944
          946    948    950    952    954    956    958    960    962    964    966    971    973    975
          977    979    981    983    985    987    989    991    993    998   1000   1002   1004   1006
         1008   1010   1012   1014   1016   1018   1020   1115   1117   1119   1121   1123   1125   1127
         1129   1131   1133   1135   1137   1139   1141   1143   1145   1147   1149   1151   1153   1155
         1157   1159   1161   1163   1165   1167   1169   1171   1173   1175   1177   1179   1182   1184
         1186   1188   1190   1192   1194   1196   1198   1200   1202   1204   1206   1208   1210   1212
         1214   1216   1218   1220   1222   1224   1226   1228   1230   1232   1234   1236   1238   1240
```

```
            1242   1244   1247   1249   1251   1253   1255   1257   1259   1261   1263   1265   1267   1269
            1271   1273   1275   1277   1279   1281   1283   1285   1287   1289   1291   1293   1295   1297
            1299   1301   1303   1305   1307   1309   1312   1314   1316   1318   1320   1322   1324   1326
            1328   1330   1332   1334   1336   1338   1340   1342   1344   1346   1348   1350   1352   1354
            1356   1358   1360   1362   1364   1366   1368   1370   1372   1374   1377   1379   1473   1475
            1477   1479   1481   1483   1485   1487   1489   1491   1493   1495   1497   1499   1501   1503
            1505   1507   1509   1511   1513   1515   1517   1519   1521   1523   1525   1527   1529   1531
            1533   1535   1537   1539   1541   1543   1545   1547   1549   1551   1553   1555   1557   1559
            1561   1563   1565   1567   1569   1571   1573   1576   1578   1672   1674   1676   1678   1680
            1682   1684   1686   1688   1690   1692   1694   1696   1698   1700   1702   1704   1706   1708
            1710   1712   1714   1716   1718   1720   1722   1724   1726   1728   1730   1732   1734   1736
            1738   1740   1742   1744   1746   1748   1750   1752   1754   1756   1758   1760   1762   1764
            1766   1768   1770   1772   1775   1777   1870   1872   1874   1876   1878   1880   1882   1884
            1886   1888   1890   1892   1894   1896   1898   1900   1902   1904   1906   1908   1910   1912
            1914   1916   1918   1920   1922   1924   1926   1928   1930   1932   1934   1936   1938   1940
            1942   1944   1946   1948   1950   1952   1954   1956   1958   1960   1962   1964   1966   1968
            1970   1973   2045   2047   2050   2052   2054   2056   2058   2061   2063   2065   2067   2070
            2072   2074   2076   2079   2081   2083   2085   2088   2090   2092   2094   2097   2099   2101
            2103   2106   2108   2110   2112   2115   2117   2119   2121   2124   2126   2128   2130   2133
            2135   2137   2139   2142   2144   2146   2148   2151   2153   2155   2157   2160   2162   2164
            2166   2169   2171   2173   2175   2178   2180   2182   2184   2187   2189   2191   2193   2196
            2198   2200   2202   2204   2207   2209   2211   2213   2215   2218   2220   2222   2224   2226
            2229   2231   2233   2235   2237   2240   2242   2244   2246   2248   2251   2253   2255   2257
            2259   2262   2264   2266   2268   2270   2273   2275   2277   2279   2281   2284   2286   2288
            2290   2292   2295   2297   2299   2301   2303   2306   2308   2310   2312   2314   2317   2319
            2321   2323   2325   2328   2330   2332   2334   2336   2339   2341   2343   2345   2347   2350
            2352   2354   2356   2358   2361   2363   2365   2367   2369   2374   2379
PNTHW        283   1108   1466   1667   1865
PNTMSG       278    281    583    586   1103   1106   1110   1461   1464   1468   1662   1665   1860   1863
PNTOCS       585
RTN           90    115    134    140    161    238    268    284    416    441    458    467    488    587
            1062   1093   1111   1420   1451   1469   1621   1652   1668   1819   1850   1866   2012   2036
SCOPER       108    262    434   1087   1445   1646   1844   2030
TMSG         586
TMSGC        277    280    583   1102   1105   1110   1460   1463   1468   1661   1664   1859   1862
```

```
                                         TITLE   DFPTA8
                                         SUBTTL  MPROC Module Tests

                                         SALL

                                         SEARCH  DFPTA,MONSYM

                                 ; INTERN's

                                         ENTRY   TSTU1,TSTU2,TSTU3,TSTU4,TSTU5,TSTU6,TSTU7,TSTU10
                                         ENTRY   TSTU11,TSTU12,TSTU13,TSTU14,TSTU15,TSTU16,TSTU17,TSTU20
                                         ENTRY   TSTU21,TSTU22,TSTU23,TSTU24,TSTU25,TSTU26,TSTU27,TSTU30
                                         ENTRY   TSTU31,TSTU32,TSTU33,TSTU34,TSTU35,TSTU36,TSTU37,TSTU40
                                         ENTRY   TSTU41

                                 ; EXTERN's

                                         EXTERN  TSTE45,TSTC3,TSTC4

                                 ; EXTERN's located in DFPTA1.MAC (Mainline code..)

                                         EXTERN  BUFF,TLOAD,CSRPNT,TRACE,TSTSUB,ODELAY,CBUF,UDEBUG
                                         EXTERN  TSLOD1,TSLOD2,CADDR,CWORDL,CWORDR,DRCRAM,DWCRAM,PARFLG,CALPAR
                                         EXTERN  AEXEC,AAPNT,ALCSR,ALFLS,PORTCI,PORTNI
                                         EXTERN  BEXEC,BBPNT,CEBUF,AEBUF
                                         EXTERN  TEXEC,TTPNT,TADDR,SEXEC,SSPNT

                                 ; EXTERN's located in DFPTAI.MAC (IPA20 Device Handling Routines module)

                                         EXTERN  LDRAR,LDCRAM,LDCSR,RDEBUF,RDLAR,RDCRAM,RDCSR
                                         EXTERN  IPACLR,IPASST,SNEXT,SLAST,SDATA
                                         EXTERN  CHINIT,CHDATA,GENCCW,INITPI,SETVEC

                                 ; EXTERN's located in DFPTAU.MAC (Utility Routines module)

                                         EXTERN  BUFGEN,SCOSW


                                 ;#*********************************************************************
                                 ; Z8 - Address for use in DDT
                                 ;#*********************************************************************

     000000'                     Z8:                                     ; address of 00000'
```

```

                                    ;#*****************************************************************
                                    ;* TEST 1 - CRAM Data Test (RAR12=0)
                                    ;
                                    ; Description:  This test verifies each CRAM location, by writing
                                    ;               a series of data patterns to and from  each  CRAM
                                    ;               location.
                                    ;
                                    ; Procedure:    Port Clear
                                    ;               Select address to write, write to RAR
                                    ;               Write data pattern to CRAM (30 bits wide)
                                    ;               Read CRAM, verify the data
                                    ;               Repeat write/read sequence with each data pattern
                                    ;
                                    ;               Repeat sequence with each CRAM location
                                    ;
                                    ; Data Patterns: zero's
                                    ;               one's
                                    ;               alternating (52..)
                                    ;               alternating (25..)
                                    ;               floating one
                                    ;               floating zero
                                    ;
                                    ; Failure:      ---
                                    ;#*****************************************************************

                                    ; Test data

000000' 254 00 0 00 000011'         TSTU1:  JRST    TG1                     ; go start test
000001' 200403  000001                      MPROC!NDMP!ZMPROC!1             ; test mask
000002' 000000  011166'                     0,,[ASCIZ ^CRAM Data Test RAR12=0^]
000003' 011173' 011201'                     [EXP M10,M18,M19,M4,M16,MLAST!M17],,[EXP MLAST!M1]
000004' 000000  000154'                     TSTU2                           ; failure test table
000005' 000000  000230'                     TSTU3                           ;   ...
000006' 000000  000305'                     TSTU4
000007' 000000  000404'                     TSTU5
000010' 777777  777777                      -1

                                    ; Start test

000011' 201 00 0 00 000000'         TG1:    MOVEI   Z8                      ; get address of module start
000012' 260 17 0 00 000000*                 GO      TRACE                   ; handle trace output
000013' 474 07 0 00 000000                  SETO    7,                      ; init CRAM address register
```

```

                                    ; First set up RAR to correct address

000014' 350 00 0 00 000007          TA1:    AOS     7                       ; point to next CRAM location
000015' 303 07 0 00 007777                  CAILE   7,7777                  ; done yet?
000016' 254 00 0 00 000050'                 JRST    TX1                     ; yes - exit
000017' 332 00 0 00 000000*                 SKIPE   UDEBUG                  ; debug mode?
                                            JRST    [CAILE  7,10            ; yes - check if exit early
                                                      JRST  TX1
000020' 254 00 0 00 011202'                           JRST  .+1]
000021' 260 17 0 00 000000*                 GO      IPACLR                  ; do a port clear
000022' 200 01 0 00 000007                  MOVE    1,7                     ; get address
000023' 242 01 0 00 000001                  LSH     1,1                     ; include bit to point to which half
000024' 202 01 0 00 011164'                 MOVEM   1,SAVCRA                ; save CRAM address
000025' 260 17 0 00 000000*                 GO      LDRAR                   ; write it to RAR register

                                    ; Next, get the data pattern to write

000026' 400 06 0 00 000000                  SETZ    6,                      ; clear bit table pointer
000027' 200 01 0 06 000053'         TB1:    MOVE    1,T1PAT(6)              ; get pattern to write
000030' 202 01 0 00 011162'                 MOVEM   1,SAVCR1                ; save data
000031' 306 01 0 00 000377                  CAIN    1,377                   ; end of table?
000032' 254 00 0 00 000014'                 JRST    TA1                     ; yes - end of segment

                                    ; Now write/read the location

000033' 350 00 0 00 000000*                 AOS     TSTSUB                  ; set up subtest number
000034' 400 15 0 00 000000          TB1L:   SETZ    ERFLG,                  ; clear error flag
000035' 260 17 0 00 000000*                 GO      LDCRAM                  ; write CRAM
000036' 260 17 0 00 000000*                 GO      RDCRAM                  ; read CRAM
000037' 202 01 0 00 011163'                 MOVEM   1,SAVCR2                ; save CRAM data
000040' 312 01 0 00 011162'                 CAME    1,SAVCR1                ; correct?
000041' 474 15 0 00 000000                  SETO    ERFLG,                  ; no - flag error
000042' 027 00 0 00 000051'                 SCOPER  MA1                     ; print error message
000043' 254 00 0 00 000034'                 JRST    TB1L                    ; loop on error
000044' 254 00 0 00 000050'                 JRST    TX1                     ; altmode exit
000045' 326 15 0 00 000050'                 JUMPN   ERFLG,TX1               ; error? yes - exit

                                    ; Increment data pattern pointer

000046' 350 00 0 00 000006                  AOS     6                       ; point to next data pattern
000047' 254 00 0 00 000027'                 JRST    TB1                     ; repeat this segment

                                    ; End of test

000050' 263 17 0 00 000000          TX1:    RTN                             ; return

                                    ; Error messages

000051' 140000  011205'             MA1:    MSG!TXNOT![ASCIZ ^CRAM write/read failed.^]
000052' 270000  011023'                     LAST!CALL!TXALL!MPNT1           ; print CRAM data correct/actual
```

```
 138
 139                                      ; Data Patterns
 140
 141  000053' 000000  000000              T1PAT:  EXP     0,7777777777,5252525252,2525252525
 142  000054' 007777  777777
 143  000055' 005252  525252
 144  000056' 002525  252525
 145  000057' 000000  000001                      EXP     0000000001,0000000002,0000000004
 146  000060' 000000  000002
 147  000061' 000000  000004
 148  000062' 000000  000010                      EXP     0000000010,0000000020,0000000040
 149  000063' 000000  000020
 150  000064' 000000  000040
 151  000065' 000000  000100                      EXP     0000000100,0000000200,0000000400
 152  000066' 000000  000200
 153  000067' 000000  000400
 154  000070' 000000  001000                      EXP     0000001000,0000002000,0000004000
 155  000071' 000000  002000
 156  000072' 000000  004000
 157  000073' 000000  010000                      EXP     0000010000,0000020000,0000040000
 158  000074' 000000  020000
 159  000075' 000000  040000
 160  000076' 000000  100000                      EXP     0000100000,0000200000,0000400000
 161  000077' 000000  200000
 162  000100' 000000  400000
 163  000101' 000001  000000                      EXP     0001000000,0002000000,0004000000
 164  000102' 000002  000000
 165  000103' 000004  000000
 166  000104' 000010  000000                      EXP     0010000000,0020000000,0040000000
 167  000105' 000020  000000
 168  000106' 000040  000000
 169  000107' 000100  000000                      EXP     0100000000,0200000000,0400000000
 170  000110' 000200  000000
 171  000111' 000400  000000
 172  000112' 001000  000000                      EXP     1000000000,2000000000,4000000000
 173  000113' 002000  000000
 174  000114' 004000  000000
 175  000115' 007777  777776                      EXP     7777777776,7777777775,7777777773
 176  000116' 007777  777775
 177  000117' 007777  777773
 178  000120' 007777  777767                      EXP     7777777767,7777777757,7777777737
 179  000121' 007777  777757
 180  000122' 007777  777737
 181  000123' 007777  777677                      EXP     7777777677,7777777577,7777777377
 182  000124' 007777  777577
 183  000125' 007777  777377
 184  000126' 007777  776777                      EXP     7777776777,7777775777,7777773777
 185  000127' 007777  775777
 186  000130' 007777  773777
 187  000131' 007777  767777                      EXP     7777767777,7777757777,7777737777
 188  000132' 007777  757777
 189  000133' 007777  737777
 190  000134' 007777  677777                      EXP     7777677777,7777577777,7777377777
 191  000135' 007777  577777
 192  000136' 007777  377777
```

```
   193  000137' 007776  777777                   EXP     7776777777,7775777777,7773777777
   194  000140' 007775  777777
   195  000141' 007773  777777
   196  000142' 007767  777777                   EXP     7767777777,7757777777,7737777777
   197  000143' 007757  777777
   198  000144' 007737  777777
   199  000145' 007677  777777                   EXP     7677777777,7577777777,7377777777
   200  000146' 007577  777777
   201  000147' 007377  777777
   202  000150' 006777  777777                   EXP     6777777777,5777777777,3777777777
   203  000151' 005777  777777
   204  000152' 003777  777777
   205  000153' 000000  000377                   EXP     377                     ; table terminator
```

```
                                     ;#*******************************************************************
                                     ;* TEST 2 - CRAM Test (RAR12=1)
                                     ;
                                     ; Description:  This test verifies each CRAM location, by writing
                                     ;               a series of data patterns to and from  each  CRAM
                                     ;               location.
                                     ;
                                     ; Procedure:    Port Clear
                                     ;               Select address to write, write to RAR
                                     ;               Write data pattern to CRAM (30 bits wide)
                                     ;               Read CRAM, verify the data
                                     ;               Repeat write/read sequence with each data pattern
                                     ;
                                     ;               Repeat sequence with each CRAM location
                                     ;
                                     ; Data Patterns: zero's
                                     ;               one's
                                     ;               alternating (52..)
                                     ;               alternating (25..)
                                     ;               floating one
                                     ;               floating zero
                                     ;
                                     ; Failure:      ---
                                     ;#*******************************************************************

                                     ; Test data

000154' 254 00 0 00 000164'          TSTU2:  JRST    TG2                     ; go start test
000155' 200403  000002                       MPROC!NDMP!ZMPROC!2             ; test mask
000156' 000000  011212'                      0,,[ASCIZ ^CRAM Data Test RAR12=1^]
000157' 011217' 011201'                      [EXP M10,M17,M19,M4,M16,MLAST!M18],,[EXP MLAST!M1]
000160' 000000  000230'                      TSTU3                           ; failure test table
000161' 000000  000305'                      TSTU4                           ;   ...
000162' 000000  000404'                      TSTU5
000163' 777777  777777                       -1

                                     ; Start test

000164' 201 00 0 00 000000'          TG2:    MOVEI   Z8                      ; get address of module start
000165' 260 17 0 00 000012*                  GO      TRACE                   ; handle trace output
000166' 474 07 0 00 000000                   SETO    7,                      ; init CRAM address register
```

```

                                         ; First set up RAR to correct address

000167' 350 00 0 00 000007       TA2:    AOS     7                       ; point to next CRAM location
000170' 303 07 0 00 007777               CAILE   7,7777                  ; done yet?
000171' 254 00 0 00 000225'              JRST    TX2                     ; yes - exit
000172' 332 00 0 00 000017*              SKIPE   UDEBUG                  ; debug mode?
                                         JRST    [CAILE  7,10            ; yes - check if exit early
                                                  JRST   TX2
000173' 254 00 0 00 011225'                       JRST   .+1]
000174' 260 17 0 00 000021*              GO      IPACLR                  ; do a port clear
000175' 200 01 0 00 000007               MOVE    1,7                     ; get address
000176' 242 01 0 00 000001               LSH     1,1                     ; include bit to point to which half
000177' 350 00 0 00 000001               AOS     1                       ; point to right half
000200' 202 01 0 00 011164'              MOVEM   1,SAVCRA                ; save CRAM address
000201' 260 17 0 00 000025*              GO      LDRAR                   ; write it to RAR register

                                 ; Next, get the data pattern to write

000202' 400 06 0 00 000000               SETZ    6,                      ; clear bit table pointer
000203' 200 01 0 06 000053'      TB2.    MOVE    1,T1PAT(6)              ; get pattern to write
000204' 202 01 0 00 011162'              MOVEM   1,SAVCR1                ; save data
000205' 306 01 0 00 000377               CAIN    1,377                   ; end of table?
000206' 254 00 0 00 000167'              JRST    TA2                     ; yes - end of segment

                                 ; Now write/read the location

000207' 350 00 0 00 000033*              AOS     TSTSUB                  ; set up subtest number
000210' 400 15 0 00 000000       TB2L:   SETZ    ERFLG,                  ; clear error flag
000211' 260 17 0 00 000035*              GO      LDCRAM                  ; write CRAM
000212' 260 17 0 00 000211*              GO      LDCRAM                  ; write CRAM
000213' 260 17 0 00 000036*              GO      RDCRAM                  ; read CRAM
000214' 202 01 0 00 011163'              MOVEM   1,SAVCR2                ; save CRAM data
000215' 312 01 0 00 011162'              CAME    1,SAVCR1                ; correct?
000216' 474 15 0 00 000000               SETO    ERFLG,                  ; no - flag error
000217' 027 00 0 00 000226'              SCOPER  MA2                     ; print error message
000220' 254 00 0 00 000210'              JRST    TB2L                    ; loop on error
000221' 254 00 0 00 000225'              JRST    TX2                     ; altmode exit
000222' 326 15 0 00 000225'              JUMPN   ERFLG,TX2               ; error? yes - exit

                                 ; Increment data pattern pointer

000223' 350 00 0 00 000006               AOS     6                       ; point to next data pattern
000224' 254 00 0 00 000203'              JRST    TB2                     ; repeat this segment

                                 ; End of test

000225' 263 17 0 00 000000       TX2:    RTN                     ; return

                                 ; Error messages

000226' 140000  011205'          MA2:    MSG!TXNOT![ASCIZ ^CRAM write/read failed.^]
000227' 270000  011023'                  LAST!CALL!TXALL!MPNT1           ; print CRAM data correct/actual
```

```
                                        ;#*********************************************************************
                                        ;* TEST 3 - CRAM Addressing Test
                                        ;
                                        ; Description:  This test  verifies  CRAM  address  selection  by
                                        ;               writing an address pattern to each location, then
                                        ;               reading each location and verifying  the  address
                                        ;               pattern read back is correct.
                                        ;
                                        ; Procedure:    Port Clear
                                        ;               For addr 0-7777:  Write RAR (left half of CRAM)
                                        ;                                 Write 30 bit address pattern
                                        ;                                 Write RAR (right half of CRAM)
                                        ;                                 Write complement of pattern
                                        ;               For addr 0-7777:  Write RAR (left half of CRAM)
                                        ;                                 Read 30 bit address pattern
                                        ;                                 Verify contents
                                        ;                                 Write RAR (right half of CRAM)
                                        ;                                 Verify contents
                                        ;
                                        ; Failure:      ---
                                        ;#*********************************************************************

                                        ; Test data

 000230' 254 00 0 00 000237'            TSTU3:  JRST    TG3                             ; go start test
 000231' 200403  000003                         MPROC!NDMP!ZMPROC!3                     ; test mask
 000232' 000000  011230'                        0,,[ASCIZ ^CRAM Addressing Test^]
 000233' 011235' 011236'                        [EXP MLAST!M4],,[EXP M16,M10,M17,M18,MLAST!M19]
 000234' 000000  000305'                        TSTU4                                   ; failure test table
 000235' 000000  000404'                        TSTU5                                   ;   ...
 000236' 777777  777777                         -1

                                        ; Start test

 000237' 201 00 0 00 000000'            TG3:    MOVEI   Z8                              ; get address of module start
 000240' 260 17 0 00 000165*                    GO      TRACE                           ; handle trace output

                                        ; Test initialization

 000241' 400 15 0 00 000000             TA3:    SETZ    ERFLG,                          ; clear error flag
 000242' 260 17 0 00 000174*                    GO      IPACLR                          ; do a port clear
 000243' 402 00 0 00 000207*                    SETZM   TSTSUB                          ; initialize subtest number
```

```

                                        ; First write the address pattern to all locations

000244' 400 07 0 00 000000                     SETZ    7,                      ; start at location 0
000245' 200 01 0 00 000007             TB3:    MOVE    1,7                     ; get address
000246' 242 01 0 00 000001                     LSH     1,1                     ; include bit to point to which half
000247' 260 17 0 00 000201*                    GO      LDRAR                   ; write it to RAR register
000250' 260 17 0 00 000212*                    GO      LDCRAM                  ; write it to CRAM (Left half)
000251' 350 00 0 00 000001                     AOS     1                       ; point to right half
000252' 260 17 0 00 000247*                    GO      LDRAR                   ; write it to RAR register
000253' 450 01 0 00 000001                     SETCA   1,1                     ; complement it
000254' 260 17 0 00 000250*                    GO      LDCRAM                  ; write it to CRAM (Right half)
000255' 350 00 0 00 000007                     AOS     7                       ; point to next CRAM location
000256' 307 07 0 00 007777                     CAIG    7,7777                  ; done yet?
000257' 254 00 0 00 000245'                    JRST    TB3                     ; no - continue

                                        ; Next read all locations and verify the result

000260' 400 07 0 00 000000                     SETZ    7,                      ; start at location 0
000261' 350 00 0 00 000243*            TC3:    AOS     TSTSUB                  ; set up subtest
000262' 200 01 0 00 000007                     MOVE    1,7                     ; get address
000263' 242 01 0 00 000001                     LSH     1,1                     ; include bit to point to which half
000264' 202 01 0 00 011164'                    MOVEM   1,SAVCRA                ; save CRAM address
000265' 202 01 0 00 011162'                    MOVEM   1,SAVCR1                ; save CRAM data (correct)
000266' 260 17 0 00 000252*                    GO      LDRAR                   ; write it to RAR register
000267' 260 17 0 00 000213*                    GO      RDCRAM2                 ; read the contents of this location
000270' 202 01 0 00 011163'                    MOVEM   1,SAVCR2                ; save CRAM data (actual)
000271' 312 01 0 00 011162'                    CAME    1,SAVCR1                ; correct?
000272' 474 15 0 00 000000                     SETO    ERFLG,                  ; no - flag error
000273' 027 00 0 00 000303'                    SCOPER  MA3                     ; print error message
000274' 254 00 0 00 000241'                    JRST    TA3                     ; loop on error
000275' 254 00 0 00 000302'                    JRST    TX3                     ; altmode exit
000276' 326 15 0 00 000302'                    JUMPN   ERFLG,TX3               ; error? yes - exit
000277' 350 00 0 00 000007                     AOS     7                       ; point to next CRAM location
000300' 307 07 0 00 007777                     CAIG    7,7777                  ; done yet?
000301' 254 00 0 00 000261'                    JRST    TC3                     ; no - continue

                                        ; End of test

000302' 263 17 0 00 000000              TX3:   RTN                     ; return

                                        ; Error messages

000303' 140000  011243'                 MA3:   MSG!TXNOT![ASCIZ ^CRAM data incorrect^]
000304' 270000  011023'                        LAST!CALL!TXALL!MPNT1           ; print CRAM data correct/actual
```

```
                                        ;#*********************************************************************
                                        ;* TEST 4 - RAR/LAR Data Path Test - LDCRAM
                                        ;
                                        ; Description:  This test verifies that the LAR  is  loaded  with
                                        ;               the contents of the RAR whenever a CRAM  location
                                        ;               is written.  This test  verifies  the  data  path
                                        ;               over the MBUS to the RAR to the LAR and back over
                                        ;               the MBUS.
                                        ;
                                        ; Procedure:    Port Clear
                                        ;               Select address to write
                                        ;               Write to RAR (left half)
                                        ;               Write data to CRAM
                                        ;               Read LAR
                                        ;               Verify that the address read is correct
                                        ;               Write to RAR (right half)
                                        ;               Write data to CRAM
                                        ;               Read LAR
                                        ;               Verify that the address read is correct
                                        ;
                                        ;               Repeat sequence with each CRAM location
                                        ;
                                        ; Failure:      ---
                                        ;#*********************************************************************

                                        ; Test data

000305' 254 00 0 00 000313'             TSTU4:  JRST    TG4                     ; go start test
000306' 210403  000004                          MPROC!MBUS!NDMP!ZMPROC!4        ; test mask
000307' 000000  011247'                         0,,[ASCIZ ^RAR/LAR Data Path Test - LDCRAM^]
000310' 011256' 011235'                         [EXP M9,MLAST!M16],,[EXP MLAST!M4]
000311' 000000  000404'                         TSTU5                           ; failure test table
000312' 777777  777777                          -1                              ;  ...

                                        ; Start test

000313' 201 00 0 00 000000'             TG4:    MOVEI   Z8                      ; get address of module start
000314' 260 17 0 00 000240*                     GO      TRACE                   ; handle trace output

                                        ; Test initialization

000315' 260 17 0 00 000242*             TA4:    GO      IPACLR                  ; do a port clear
000316' 201 01 0 00 040000                      MOVEI   1,SELLAR                ; set this bit
000317' 260 17 0 00 000000*                     GO      LDCSR                   ; write CSR register
000320' 400 07 0 00 000000                      SETZ    7,                      ; start at location 0
```

```

                                  ; Write address, read LAR, verify result (left half)

000321' 400 15 0 00 000000        TB4:    SETZ    ERFLG,                  ; clear error flag
000322' 200 01 0 00 000007                MOVE    1,7                     ; get address
000323' 242 01 0 00 000001                LSH     1,1                     ; include bit to point to which half
000324' 202 01 0 00 011162'               MOVEM   1,SAVCR1                ; save CRAM address
000325' 202 01 0 00 000261*               MOVEM   1,TSTSUB                ; set up subtest number
000326' 350 00 0 00 000325*               AOS     TSTSUB                  ;   ...
000327' 260 17 0 00 000266*               GO      LDRAR                   ; write it to RAR register
000330' 450 01 0 00 000001                SETCA   1,1                     ; complement the data
000331' 260 17 0 00 000254*               GO      LDCRAM                  ; write to CRAM
000332' 260 17 0 00 000000*               GO      RDLAR                   ; read LAR
000333' 202 01 0 00 011163'               MOVEM   1,SAVCR2                ; save data read
000334' 312 01 0 00 011162'               CAME    1,SAVCR1                ; correct?
000335' 474 15 0 00 000000                SETO    ERFLG,                  ; no - flag error
000336' 027 00 0 00 000370'               SCOPER  MA4                     ; print error message
000337' 254 00 0 00 000321'               JRST    TB4                     ; loop on error
000340' 254 00 0 00 000367'               JRST    TX4                     ; altmode exit
000341' 326 15 0 00 000367'               JUMPN   ERFLG,TX4               ; error? yes - exit

                                  ; Write address, read LAR, verify result (right half)

000342' 400 15 0 00 000000        TC4:    SETZ    ERFLG,                  ; clear error flag
000343' 200 01 0 00 000007                MOVE    1,7                     ; get address
000344' 242 01 0 00 000001                LSH     1,1                     ; include bit to point to which half
000345' 350 00 0 00 000001                AOS     1                       ; point to right half
000346' 202 01 0 00 011162'               MOVEM   1,SAVCR1                ; save CRAM address
000347' 202 01 0 00 000326*               MOVEM   1,TSTSUB                ; set up subtest number
000350' 350 00 0 00 000347*               AOS     TSTSUB                  ;   ...
000351' 260 17 0 00 000327*               GO      LDRAR                   ; write it to RAR register
000352' 450 01 0 00 000001                SETCA   1,1                     ; complement the data
000353' 260 17 0 00 000331*               GO      LDCRAM                  ; write to CRAM
000354' 260 17 0 00 000332*               GO      RDLAR                   ; read LAR
000355' 202 01 0 00 011163'               MOVEM   1,SAVCR2                ; save data read
000356' 312 01 0 00 011162'               CAME    1,SAVCR1                ; correct?
000357' 474 15 0 00 000000                SETO    ERFLG,                  ; no - flag error
000360' 027 00 0 00 000370'               SCOPER  MA4                     ; print error message
000361' 254 00 0 00 000342'               JRST    TC4                     ; loop on error
000362' 254 00 0 00 000367'               JRST    TX4                     ; altmode exit
000363' 326 15 0 00 000367'               JUMPN   ERFLG,TX4               ; error? yes - exit

                                  ; Point to next address

000364' 350 00 0 00 000007                AOS     7                       ; point to next CRAM location
000365' 307 07 0 00 007777                CAIG    7,7777                  ; done yet?
000366' 254 00 0 00 000321'               JRST    TB4                     ; no - continue

                                  ; End of test

000367' 263 17 0 00 000000        TX4:    RTN                     ; return
```

```



                                       ; Error messages

000370' 140000  011260'                MA4:    MSG!TXNOT![ASCIZ ^LAR not loaded correctly by LDCRAM of location ^]
000371' 240000  000373'                        CALL!TXNOT!MA4PNT
000372' 270000  011052'                        LAST!CALL!TXALL!MPNT2           ; print LAR data correct/actual

                                       ; Print CRAM address written

000373' 200 00 0 00 011162'            MA4PNT: MOVE    SAVCR1
000374' 242 01 0 00 000001                     LSH     1,1
000375' 037 04 0 00 000000                     PNT4
000376' 200 00 0 00 011162'                    MOVE    SAVCR1
000377' 606 00 0 00 000001                     TRNN    1
000400' 037 00 0 00 011272'                    TMSG    < (Left)>
000401' 602 00 0 00 000001                     TRNE    1
000402' 037 00 0 00 011274'                    TMSG    < (Right)>
000403' 263 17 0 00 000000                     RTN
```

```

                                    ;#*********************************************************************
                                    ;* TEST 5 - RAR/LAR Data Path Test - RDCRAM
                                    ;
                                    ; Description:  This test verifies that the LAR  is  loaded  with
                                    ;               the contents of the RAR whenever a CRAM  location
                                    ;               is read.  This test verifies the  data  path over
                                    ;               the MBUS to the RAR to the LAR and back over  the
                                    ;               MBUS.
                                    ;
                                    ; Procedure:    Port Clear
                                    ;               Select address to write
                                    ;               Write RAR (left half)
                                    ;               Read CRAM
                                    ;               Read LAR
                                    ;               Verify that the address read is correct
                                    ;               Write RAR (right half)
                                    ;               Read CRAM
                                    ;               Read LAR
                                    ;               Verify that the address read is correct
                                    ;
                                    ;               Repeat sequence with each CRAM location
                                    ;
                                    ; Failure:      ---
                                    ;#*********************************************************************

                                    ; Test data

 000404' 254 00 0 00 000411'        TSTU5:  JRST    TG5                     ; go start test
 000405' 210403  000005                     MPROC!MBUS!NDMP!ZMPROC!5        ; test mask
 000406' 000000  011276'                    0,,[ASCIZ ^RAR/LAR Data Path Test - RDCRAM^]
 000407' 011256' 011235'                    [EXP M9,MLAST!M16],,[EXP MLAST!M4]
 000410' 777777  777777                     -1                              ; failure test table

                                    ; Start test

 000411' 201 00 0 00 000000'        TG5:    MOVEI   Z8                      ; get address of module start
 000412' 260 17 0 00 000314*                GO      TRACE                   ; handle trace output

                                    ; Test initialization

 000413' 260 17 0 00 000315*        TA5:    GO      IPACLR                  ; do a port clear
 000414' 201 01 0 00 040000                 MOVEI   1,SELLAR                ; set this bit
 000415' 260 17 0 00 000317*                GO      LDCSR                   ; write CSR register
 000416' 400 07 0 00 000000                 SETZ    7,                      ; start at location 0
```

```

                                        ; Write address, read LAR, verify result (left half)

000417' 400 15 0 00 000000      TB5:    SETZ    ERFLG,                  ; clear error flag
000420' 200 01 0 00 000007              MOVE    1,7                     ; get address
000421' 242 01 0 00 000001              LSH     1,1                     ; include bit to point to which half
000422' 202 01 0 00 011162'             MOVEM   1,SAVCR1                ; save CRAM address
000423' 202 01 0 00 000350*             MOVEM   1,TSTSUB                ; set up subtest number
000424' 350 00 0 00 000423*             AOS     TSTSUB                  ;   ...
000425' 260 17 0 00 000351*             GO      LDRAR                   ; write it to RAR register
000426' 400 01 0 00 000000              SETZ    1,                      ; set up CSR to read CRAM
000427' 260 17 0 00 000415*             GO      LDCSR                   ; write CSR register
000430' 260 17 0 00 000267*             GO      RDCRAM                  ; read CRAM
000431' 201 01 0 00 040000              MOVEI   1,SELLAR                ; set this bit
000432' 260 17 0 00 000427*             GO      LDCSR                   ; write CSR register
000433' 260 17 0 00 000354*             GO      RDLAR                   ; read LAR
000434' 202 01 0 00 011163'             MOVEM   1,SAVCR2                ; save data read
000435' 312 01 0 00 011162'             CAME    1,SAVCR1                ; correct?
000436' 474 15 0 00 000000              SETO    ERFLG,                  ; no - flag error
000437' 027 00 0 00 000474'             SCOPER  MA5                     ; print error message
000440' 254 00 0 00 000417'             JRST    TB5                     ; loop on error
000441' 254 00 0 00 000473'             JRST    TX5                     ; altmode exit
000442' 326 15 0 00 000473'             JUMPN   ERFLG,TX5               ; error? yes - exit

                                        ; Write address, read LAR, verify result (right half)

000443' 400 15 0 00 000000      TC5:    SETZ    ERFLG,                  ; clear error flag
000444' 200 01 0 00 000007              MOVE    1,7                     ; get address
000445' 242 01 0 00 000001              LSH     1,1                     ; include bit to point to which half
000446' 350 00 0 00 000001              AOS     1                       ; point to right half
000447' 202 01 0 00 011162'             MOVEM   1,SAVCR1                ; save CRAM address
000450' 202 01 0 00 000424*             MOVEM   1,TSTSUB                ; set up subtest number
000451' 350 00 0 00 000450*             AOS     TSTSUB                  ;   ...
000452' 260 17 0 00 000425*             GO      LDRAR                   ; write it to RAR register
000453' 400 01 0 00 000000              SETZ    1,                      ; set up CSR to read CRAM
000454' 260 17 0 00 000432*             GO      LDCSR                   ; write CSR register
000455' 260 17 0 00 000430*             GO      RDCRAM                  ; read CRAM
000456' 201 01 0 00 040000              MOVEI   1,SELLAR                ; set this bit
000457' 260 17 0 00 000454*             GO      LDCSR                   ; write CSR register
000460' 260 17 0 00 000433*             GO      RDLAR                   ; read LAR
000461' 202 01 0 00 011163'             MOVEM   1,SAVCR2                ; save data read
000462' 312 01 0 00 011162'             CAME    1,SAVCR1                ; correct?
000463' 474 15 0 00 000000              SETO    ERFLG,                  ; no - flag error
000464' 027 00 0 00 000474'             SCOPER  MA5                     ; print error message
000465' 254 00 0 00 000443'             JRST    TC5                     ; loop on error
000466' 254 00 0 00 000473'             JRST    TX5                     ; altmode exit
000467' 326 15 0 00 000473'             JUMPN   ERFLG,TX5               ; error? yes - exit
```

```

                                       ; Point to next address

 000470' 350 00 0 00 000007                    AOS     7                               ; point to next CRAM location
 000471' 307 07 0 00 007777                    CAIG    7,7777                          ; done yet?
 000472' 254 00 0 00 000417'                   JRST    TB5                             ; no - continue

                                       ; End of test

 000473' 263 17 0 00 000000            TX5:    RTN                     ; return

                                       ; Error messages

 000474' 140000  011305'               MA5:    MSG!TXNOT![ASCIZ ^LAR not loaded correctly by RDCRAM of location ^]
 000475' 240000  000477'                       CALL!TXNOT!MA5PNT
 000476' 270000  011052'                       LAST!CALL!TXALL!MPNT2           ; print LAR data correct/actual

                                       ; Print CRAM address written

 000477' 200 00 0 00 011162'           MA5PNT: MOVE    SAVCR1
 000500' 242 01 0 00 000001                    LSH     1,1
 000501' 037 04 0 00 000000                    PNT4
 000502' 200 00 0 00 011162'                   MOVE    SAVCR1
 000503' 606 00 0 00 000001                    TRNN    1
 000504' 037 00 0 00 011272'                   TMSG    < (Left)>
 000505' 602 00 0 00 000001                    TRNE    1
 000506' 037 00 0 00 011274'                   TMSG    < (Right)>
 000507' 263 17 0 00 000000                    RTN
```

```

                                       ;#******************************************************************
                                       ;* TEST 6 - Cram Ctrl Register Test
                                       ;
                                       ; Description:  Verify each bit in the CRAM Control  Register  by
                                       ;               writing a 0 or 1 to it and verifying that a  CRAM
                                       ;               parity error does not occur.
                                       ;
                                       ; Procedure:    Port Clear
                                       ;               Load CRAM locations 0-59. with a floating 1 (good
                                       ;                 parity).
                                       ;               Single step a microinstruction at  each  location
                                       ;                 verifying that a CRAM PE does not occur.
                                       ;
                                       ;               Do for locations 0-73 (octal).
                                       ;
                                       ; Failure:      ---
                                       ;#******************************************************************

                                       ; Test data

000510' 254 00 0 00 000520'            TSTU6:  JRST    TG6                     ; go start test
000511' 200403  000006                         MPROC!NDMP!ZMPROC!6             ; test mask
000512' 000000  011317'                        0,,[ASCIZ ^Cram Ctrl Register Test^]
000513' 011324' 011325'                        [EXP MLAST!M12],,[EXP MLAST!E15]
000514' 000000  001017'                        TSTU7                           ; failure test table
000515' 000000000000#                          TSTE45                          ;   ...
000516' 000000  010671'                        TSTU41
000517' 777777  777777                         -1

                                       ; Start test

000520' 201 00 0 00 000000'            TG6:    MOVEI   Z8                      ; get address of module start
000521' 260 17 0 00 000412*                    GO      TRACE                   ; handle trace output
000522' 260 17 0 00 000413*                    GO      IPACLR                  ; do a 'port clear'

                                       ; Load microcode (floating 1 in each location - with good parity)

000523' 201 06 0 00 000625'                    MOVEI   6,T6TAB                 ; get address of table
000524' 476 00 0 00 000000*                    SETOM   CADDR                   ; initialize CRAM address
000525' 350 00 0 00 000524*            TI6A:   AOS     CADDR                   ; increment CRAM address
000526' 120 02 0 06 000000                     DMOVE   2,(6)                   ; get microword data
000527' 306 02 0 00 000377                     CAIN    2,377                   ; end of table?
000530' 254 00 0 00 000541'                    JRST    TL6                     ; yes - continue
000531' 202 02 0 00 000000*                    MOVEM   2,CWORDL                ; save left 30 bits
000532' 202 03 0 00 000000*                    MOVEM   3,CWORDR                ; save right 30 bits
000533' 261 17 0 00 000000*                    PUT     PARFLG                  ; save state of parity flag
000534' 476 00 0 00 000533*                    SETOM   PARFLG                  ; ensure correct parity is generated
000535' 260 17 0 00 000000*                    GO      DWCRAM                  ; write the data
000536' 262 17 0 00 000534*                    GET     PARFLG                  ; restore state of parity flag
000537' 271 06 0 00 000002                     ADDI    6,2                     ; poin to next entry
000540' 254 00 0 00 000525'                    JRST    TI6A                    ; loop till done
```

```

                                        ; Initialization

000541' 400 10 0 00 000000      TL6:    SETZ    10,                     ; initialize start address
000542' 201 00 0 00 000010              MOVEI   MPRUN                   ; initialize start data
000543' 202 00 0 00 000000*             MOVEM   SDATA

                                        ; Loop on start address

000544' 400 15 0 00 000000      TA6:    SETZ    ERFLG,                  ; clear error flag
000545' 201 00 0 10 000001              MOVEI   1(10)                   ; set up subtest number
000546' 202 00 0 00 000451*             MOVEM   TSTSUB                  ;   ...
000547' 260 17 0 00 000522*             GO      IPACLR                  ; clear port
000550' 202 10 0 00 000000*             MOVEM   10,SNEXT                ; set up start data
000551' 260 17 0 00 000000*             GO      IPASST                  ; single step once
000552' 200 15 0 00 011326'             MOVE    ERFLG,[-2]              ; error - reading/writing CSR
000553' 200 15 0 00 011326'             MOVE    ERFLG,[-2]              ; error - port already running
000554' 200 15 0 00 011326'             MOVE    ERFLG,[-2]              ; error - error bits set in CSR
000555' 326 15 0 00 000562'             JUMPN   ERFLG,TA6A              ; error yet - yes - continue
000556' 260 17 0 00 000000*             GO      RDCSR                   ; read the CSR register
000557' 474 15 0 00 000000              SETO    ERFLG,                  ; error
000560' 603 01 0 00 004000              TLNE    1,(CRAMPE)              ; get a CRAM PE?
000561' 474 15 0 00 000000              SETO    ERFLG,                  ; yes - set error flag
000562' 027 00 0 00 000572'     TA6A:   SCOPER  MA6                     ; print error message
000563' 254 00 0 00 000544'             JRST    TA6                     ; loop on error
000564' 254 00 0 00 000571'             JRST    TX6                     ; altmode exit
000565' 326 15 0 00 000571'             JUMPN   ERFLG,TX6               ; error yet? - yes - exit test

                                        ; Next address

000566' 350 00 0 00 000010              AOS     10                      ; point to next address
000567' 307 10 0 00 000073              CAIG    10,73                   ; done yet?
000570' 254 00 0 00 000544'             JRST    TA6                     ; no - loop till done

                                        ; End of test

000571' 263 17 0 00 000000      TX6:    RTN                             ; return

                                        ; Error messages

000572' 270000  000573'         MA6:    LAST!CALL!TXALL!MA6PNT          ; print error data

000573' 311 15 0 00 011327'     MA6PNT: CAML    ERFLG,[-1]              ; start up error?
000574' 254 00 0 00 000606'             JRST    MA6PN0                  ; no - print other message
000575' 200 00 0 00 000000*             MOVE    SCOSW                   ; get switches
000576' 607 00 0 00 000200              TLNN    TXTINH                  ; text inhibit switch set?
000577' 037 00 0 00 011330*             TMSGC   <Couldn't single step the port>
000600' 260 17 0 00 000556*             GO      RDCSR                   ; get CSR contents
000601' 255 00 0 00 000000              JFCL                            ; error
000602' 201 02 0 00 000023              MOVEI   2,^D19                  ; character count so far
000603' 201 03 0 00 000023              MOVEI   3,^D19                  ; continuation line column number
000604' 260 17 0 00 000000*             GO      CSRPNT                  ; go print in English
000605' 263 17 0 00 000000              RTN
```

```

000606' 200 00 0 00 000575*     MA6PN0: MOVE    SCOSW                   ; get switches
000607' 603 00 0 00 000200              TLNE    TXTINH                  ; text inhibit switch set?
                                        JRST    [TMSGC  <CRAM Loc >     ; yes - only print the CRAM
                                                 MOVE   SLAST           ;   address
                                                 PNT4
000610' 254 00 0 00 011342'                      RTN]
000611' 037 00 0 00 011346'             TMSGC   <CRAM PE occurred when single stepping location >
000612' 200 00 0 00 000000*             MOVE    SLAST
000613' 037 04 0 00 000000              PNT4
000614' 037 00 0 00 011337'             TMSGC   <CRAM Loc >
000615' 200 00 0 00 000612*             MOVE    SLAST
000616' 037 04 0 00 000000              PNT4
000617' 037 00 0 00 011360'             TMSG    < contained a 1 in bit >
000620' 201 00 0 00 000073              MOVEI   ^D59
000621' 274 00 0 00 000615*             SUB     SLAST
000622' 037 15 0 00 000000              PNTDEC
000623' 037 00 0 00 011365'             TMSG    <.>
000624' 263 17 0 00 000000              RTN

                                ; Microcode:

000625' 000000  400000          T6TAB:  EXP     1B18!0,1                ; 0
000626' 000000  000001
000627' 000000  000000                  EXP     0,2                     ; 1
000630' 000000  000002
000631' 000000  000000                  EXP     0,4                     ; 2
000632' 000000  000004
000633' 000000  000000                  EXP     0,10                    ; 3
000634' 000000  000010
000635' 000000  000000                  EXP     0,20                    ; 4
000636' 000000  000020
000637' 000000  000000                  EXP     0,40                    ; 5
000640' 000000  000040
000641' 000000  000000                  EXP     0,100                   ; 6
000642' 000000  000100
000643' 000000  000000                  EXP     0,200                   ; 7
000644' 000000  000200
000645' 000000  000000                  EXP     0,400                   ; 10
000646' 000000  000400
000647' 000000  000000                  EXP     0,1000                  ; 11
000650' 000000  001000
000651' 000000  000000                  EXP     0,2000                  ; 12
000652' 000000  002000
000653' 000000  000000                  EXP     0,4000                  ; 13
000654' 000000  004000
000655' 000000  000000                  EXP     0,10000                 ; 14
000656' 000000  010000
000657' 000000  000000                  EXP     0,20000                 ; 15
000660' 000000  020000
000661' 000000  000000                  EXP     0,40000                 ; 16
000662' 000000  040000
000663' 000000  000000                  EXP     0,100000                ; 17
000664' 000000  100000
000665' 000000  000000                  EXP     0,200000                ; 20
```

```
 784  000666' 000000  200000
 785  000667' 000000  000000                    EXP     0,400000                ; 21
 786  000670' 000000  400000
 787  000671' 000000  000000                    EXP     0,1000000               ; 22
 788  000672' 000001  000000
 789  000673' 000000  000000                    EXP     0,2000000               ; 23
 790  000674' 000002  000000
 791  000675' 000000  000000                    EXP     0,4000000               ; 24
 792  000676' 000004  000000
 793  000677' 000000  000000                    EXP     0,10000000              ; 25
 794  000700' 000010  000000
 795  000701' 000000  000000                    EXP     0,20000000              ; 26
 796  000702' 000020  000000
 797  000703' 000000  000000                    EXP     0,40000000              ; 27
 798  000704' 000040  000000
 799  000705' 000000  000000                    EXP     0,100000000             ; 30
 800  000706' 000100  000000
 801  000707' 000000  000000                    EXP     0,200000000             ; 31
 802  000710' 000200  000000
 803  000711' 000000  000000                    EXP     0,400000000             ; 32
 804  000712' 000400  000000
 805  000713' 000000  000000                    EXP     0,1000000000            ; 33
 806  000714' 001000  000000
 807  000715' 000000  000000                    EXP     0,2000000000            ; 34
 808  000716' 002000  000000
 809  000717' 000000  000000                    EXP     0,4000000000            ; 35
 810  000720' 004000  000000
 811  000721' 000000  000000                    EXP     0,1                     ; 36
 812  000722' 000000  000001
 813  000723' 000000  000000                    EXP     0,2                     ; 37
 814  000724' 000000  000002
 815  000725' 000000  000000                    EXP     0,4                     ; 40
 816  000726' 000000  000004
 817  000727' 000000  000000                    EXP     0,10                    ; 41
 818  000730' 000000  000010
 819  000731' 000000  000000                    EXP     0,20                    ; 42
 820  000732' 000000  000020
 821  000733' 000000  000000                    EXP     0,40                    ; 43
 822  000734' 000000  000040
 823  000735' 000000  000000                    EXP     0,100                   ; 44
 824  000736' 000000  000100
 825  000737' 000000  000000                    EXP     0,200                   ; 45
 826  000740' 000000  000200
 827  000741' 000000  000000                    EXP     0,400                   ; 46
 828  000742' 000000  000400
 829  000743' 000000  000000                    EXP     0,1000                  ; 47
 830  000744' 000000  001000
 831  000745' 000000  000000                    EXP     0,2000                  ; 50
 832  000746' 000000  002000
 833  000747' 000000  000000                    EXP     0,4000                  ; 51
 834  000750' 000000  004000
 835  000751' 000000  000000                    EXP     0,10000                 ; 52
 836  000752' 000000  010000
 837  000753' 000000  000000                    EXP     0,20000                 ; 53
 838  000754' 000000  020000
 839 000755' 000000  000000            EXP     0,40000                 ; 54
 840 000756' 000000  040000
 841 000757' 000000  000000            EXP     0,100000                ; 55
 842 000760' 000000  100000
 843 000761' 000000  000000            EXP     0,200000                ; 56
 844 000762' 000000  200000
 845 000763' 000000  000000            EXP     0,400000                ; 57
 846 000764' 000000  400000
 847 000765' 000000  000000            EXP     0,1000000               ; 60
 848 000766' 000001  000000
 849 000767' 000000  000000            EXP     0,2000000               ; 61
 850 000770' 000002  000000
 851 000771' 000000  000000            EXP     0,4000000               ; 62
 852 000772' 000004  000000
 853 000773' 000000  000000            EXP     0,10000000              ; 63
 854 000774' 000010  000000
 855 000775' 000000  000000            EXP     0,20000000              ; 64
 856 000776' 000020  000000
 857 000777' 000000  000000            EXP     0,40000000              ; 65
 858 001000' 000040  000000
 859 001001' 000000  000000            EXP     0,100000000             ; 66
 860 001002' 000100  000000
 861 001003' 000000  000000            EXP     0,200000000             ; 67
 862 001004' 000200  000000
 863 001005' 000000  000000            EXP     0,400000000             ; 70
 864 001006' 000400  000000
 865 001007' 000000  000000            EXP     0,1000000000            ; 71
 866 001010' 001000  000000
 867 001011' 000000  000000            EXP     0,2000000000            ; 72
 868 001012' 002000  000000
 869 001013' 000000  000000            EXP     0,4000000000            ; 73
 870 001014' 004000  000000
 871 001015' 000000  000377            EXP     377,377
 872 001016' 000000  000377
```

```
                                              ;#*****************************************************************
                                              ;* TEST 7 - Cram Ctrl Register Test
                                              ;
                                              ; Description:  Verify each bit in the CRAM Control  Register  by
                                              ;               writing a 0 or 1 to it and verifying that a  CRAM
                                              ;               parity error does occur, when that word also  has
                                              ;               bad parity.
                                              ;
                                              ; Procedure:    Port Clear
                                              ;               Load CRAM locations 0-59. with a  floating 1 (bad
                                              ;                 parity).
                                              ;               Single step a microinstruction at  each  location
                                              ;                 verifying that a CRAM PE does occur.
                                              ;
                                              ;               Do for locations 0-73 (octal).
                                              ;
                                              ; Failure:      ---
                                              ;#*****************************************************************

                                              ; Test data

001017' 254 00 0 00 001026'                   TSTU7:  JRST    TG7                     ; go start test
001020' 200403  000007                                MPROC!NDMP!ZMPROC!7             ; test mask
001021' 000000  011317'                               0,,[ASCIZ ^Cram Ctrl Register Test^]
001022' 011324' 011366'                               [EXP MLAST!M12],,[EXP E15,MLAST!E16]
001023' 000000000000#                                 TSTE45                          ; failure test table
001024' 000000  010671'                               TSTU41                          ;   ...
001025' 777777  777777                                -1

                                              ; Start test

001026' 201 00 0 00 000000'                   TG7:    MOVEI   Z8                      ; get address of module start
001027' 260 17 0 00 000521*                           GO      TRACE                   ; handle trace output
001030' 260 17 0 00 000547*                           GO      IPACLR                  ; do a 'port clear'

                                              ; Load microcode (floating 1 in each location - with good parity)

001031' 201 06 0 00 001133'                           MOVEI   6,T7TAB                 ; get address of table
001032' 476 00 0 00 000525*                           SETOM   CADDR                   ; initialize CRAM address
001033' 350 00 0 00 001032*                   TI7A:   AOS     CADDR                   ; increment CRAM address
001034' 120 02 0 06 000000                            DMOVE   2,(6)                   ; get microword data
001035' 306 02 0 00 000377                            CAIN    2,377                   ; end of table?
001036' 254 00 0 00 001047'                           JRST    TL7                     ; yes - continue
001037' 202 02 0 00 000531*                           MOVEM   2,CWORDL                ; save left 30 bits
001040' 202 03 0 00 000532*                           MOVEM   3,CWORDR                ; save right 30 bits
001041' 261 17 0 00 000536*                           PUT     PARFLG                  ; save state of parity flag
001042' 402 00 0 00 001041*                           SETZM   PARFLG                  ; set to not calculate parity
001043' 260 17 0 00 000535*                           GO      DWCRAM                  ; write the data
001044' 262 17 0 00 001042*                           GET     PARFLG                  ; restore state of parity flag
001045' 271 06 0 00 000002                            ADDI    6,2                     ; poin to next entry
001046' 254 00 0 00 001033'                           JRST    TI7A                    ; loop till done
```

```


                                   ; Initialization

001047' 400 10 0 00 000000         TL7:    SETZ    10,                     ; initialize start address
001050' 201 00 0 00 000010                 MOVEI   MPRUN                   ; initialize start data
001051' 202 00 0 00 000543*                MOVEM   SDATA

                                   ; Loop on start address

001052' 400 15 0 00 000000         TA7:    SETZ    ERFLG,                  ; clear error flag
001053' 201 00 0 10 000001                 MOVEI   1(10)                   ; set up subtest number
001054' 202 00 0 00 000546*                MOVEM   TSTSUB                  ;   ...
001055' 260 17 0 00 001030*                GO      IPACLR                  ; clear port
001056' 202 10 0 00 000550*                MOVEM   10,SNEXT                ; set up start data
001057' 260 17 0 00 000551*                GO      IPASST                  ; single step once
001060' 200 15 0 00 011326'                MOVE    ERFLG,[-2]              ; error - reading/writing CSR
001061' 200 15 0 00 011326'                MOVE    ERFLG,[-2]              ; error - port already running
001062' 200 15 0 00 011326'                MOVE    ERFLG,[-2]              ; error - error bits set in CSR
001063' 326 15 0 00 001070'                JUMPN   ERFLG,TA7A              ; error yet - yes - continue
001064' 260 17 0 00 000600*                GO      RDCSR                   ; read the CSR register
001065' 474 15 0 00 000000                 SETO    ERFLG,                  ; error
001066' 607 01 0 00 004000                 TLNN    1,(CRAMPE)              ; get a CRAM PE?
001067' 474 15 0 00 000000                 SETO    ERFLG,                  ; no - set error flag
001070' 027 00 0 00 001100'        TA7A:   SCOPER  MA7                     ; print error message
001071' 254 00 0 00 001052'                JRST    TA7                     ; loop on error
001072' 254 00 0 00 001077'                JRST    TX7                     ; altmode exit
001073' 326 15 0 00 001077'                JUMPN   ERFLG,TX7               ; error yet? - yes - exit test

                                   ; Next address

001074' 350 00 0 00 000010                 AOS     10                      ; point to next address
001075' 307 10 0 00 000073                 CAIG    10,73                   ; done yet?
001076' 254 00 0 00 001052'                JRST    TA7                     ; no - loop till done

                                   ; End of test

001077' 263 17 0 00 000000         TX7:    RTN                             ; return

                                   ; Error messages

001100' 270000  001101'            MA7:    LAST!CALL!TXALL!MA7PNT          ; print error data

001101' 311 15 0 00 011327'        MA7PNT: CAML    ERFLG,[-1]              ; start up error?
001102' 254 00 0 00 001114'                JRST    MA7PN0                  ; no - print other message
001103' 200 00 0 00 000606*                MOVE    SCOSW                   ; get switches
001104' 607 00 0 00 000200                 TLNN    TXTINH                  ; text inhibit switch set?
001105' 037 00 0 00 011330'                TMSGC   <Couldn't single step the port>
001106' 260 17 0 00 001064*                GO      RDCSR                   ; get CSR contents
001107' 255 00 0 00 000000                 JFCL                            ; error
001110' 201 02 0 00 000023                 MOVEI   2,^D19                  ; character count so far
001111' 201 03 0 00 000023                 MOVEI   3,^D19                  ; continuation line column number
001112' 260 17 0 00 000604*                GO      CSRPNT                  ; go print in English
001113' 263 17 0 00 000000                 RTN
```

```

001114' 200 00 0 00 001103*     MA7PN0: MOVE    SCOSW                   ; get switches
001115' 603 00 0 00 000200              TLNE    TXTINH                  ; text inhibit switch set?
                                        JRST    [TMSGC  <CRAM Loc >     ; yes - only print the CRAM
                                                 MOVE   SLAST           ;   address
                                                 PNT4
001116' 254 00 0 00 011342'                      RTN]
001117' 037 00 0 00 011370'             TMSGC   <CRAM PE did not occur when single stepping location >
001120' 200 00 0 00 000621*             MOVE    SLAST
001121' 037 04 0 00 000000              PNT4
001122' 037 00 0 00 011337'             TMSGC   <CRAM Loc >
001123' 200 00 0 00 001120*             MOVE    SLAST
001124' 037 04 0 00 000000              PNT4
001125' 037 00 0 00 011360'             TMSG    < contained a 1 in bit >
001126' 201 00 0 00 000073              MOVEI   ^D59
001127' 274 00 0 00 001123*             SUB     SLAST
001130' 037 15 0 00 000000              PNTDEC
001131' 037 00 0 00 011365'             TMSG    <.>
001132' 263 17 0 00 000000              RTN

                                ; Microcode:

001133' 000000  000000          T7TAB:  EXP     0,1                     ; 0
001134' 000000  000001
001135' 000000  400000                  EXP     1B18!0,2                ; 1
001136' 000000  000002
001137' 000000  400000                  EXP     1B18!0,4                ; 2
001140' 000000  000004
001141' 000000  400000                  EXP     1B18!0,10               ; 3
001142' 000000  000010
001143' 000000  400000                  EXP     1B18!0,20               ; 4
001144' 000000  000020
001145' 000000  400000                  EXP     1B18!0,40               ; 5
001146' 000000  000040
001147' 000000  400000                  EXP     1B18!0,100              ; 6
001150' 000000  000100
001151' 000000  400000                  EXP     1B18!0,200              ; 7
001152' 000000  000200
001153' 000000  400000                  EXP     1B18!0,400              ; 10
001154' 000000  000400
001155' 000000  400000                  EXP     1B18!0,1000             ; 11
001156' 000000  001000
001157' 000000  400000                  EXP     1B18!0,2000             ; 12
001160' 000000  002000
001161' 000000  400000                  EXP     1B18!0,4000             ; 13
001162' 000000  004000
001163' 000000  400000                  EXP     1B18!0,10000            ; 14
001164' 000000  010000
001165' 000000  400000                  EXP     1B18!0,20000            ; 15
001166' 000000  020000
001167' 000000  400000                  EXP     1B18!0,40000            ; 16
001170' 000000  040000
001171' 000000  400000                  EXP     1B18!0,100000           ; 17
001172' 000000  100000
001173' 000000  400000                  EXP     1B18!0,200000           ; 20
```

```
1033  001174' 000000  200000
1034  001175' 000000  400000          EXP     1B18!0,400000           ; 21
1035  001176' 000000  400000
1036  001177' 000000  400000          EXP     1B18!0,1000000          ; 22
1037  001200' 000001  000000
1038  001201' 000000  400000          EXP     1B18!0,2000000          ; 23
1039  001202' 000002  000000
1040  001203' 000000  400000          EXP     1B18!0,4000000          ; 24
1041  001204' 000004  000000
1042  001205' 000000  400000          EXP     1B18!0,10000000         ; 25
1043  001206' 000010  000000
1044  001207' 000000  400000          EXP     1B18!0,20000000         ; 26
1045  001210' 000020  000000
1046  001211' 000000  400000          EXP     1B18!0,40000000         ; 27
1047  001212' 000040  000000
1048  001213' 000000  400000          EXP     1B18!0,100000000        ; 30
1049  001214' 000100  000000
1050  001215' 000000  400000          EXP     1B18!0,200000000        ; 31
1051  001216' 000200  000000
1052  001217' 000000  400000          EXP     1B18!0,400000000        ; 32
1053  001220' 000400  000000
1054  001221' 000000  400000          EXP     1B18!0,1000000000       ; 33
1055  001222' 001000  000000
1056  001223' 000000  400000          EXP     1B18!0,2000000000       ; 34
1057  001224' 002000  000000
1058  001225' 000000  400000          EXP     1B18!0,4000000000       ; 35
1059  001226' 004000  000000
1060  001227' 000000  400001          EXP     1B18!1,0                ; 36
1061  001230' 000000  000000
1062  001231' 000000  400002          EXP     1B18!2,0                ; 37
1063  001232' 000000  000000
1064  001233' 000000  400004          EXP     1B18!4,0                ; 40
1065  001234' 000000  000000
1066  001235' 000000  400010          EXP     1B18!10,0               ; 41
1067  001236' 000000  000000
1068  001237' 000000  400020          EXP     1B18!20,0               ; 42
1069  001240' 000000  000000
1070  001241' 000000  400040          EXP     1B18!40,0               ; 43
1071  001242' 000000  000000
1072  001243' 000000  400100          EXP     1B18!100,0              ; 44
1073  001244' 000000  000000
1074  001245' 000000  400200          EXP     1B18!200,0              ; 45
1075  001246' 000000  000000
1076  001247' 000000  400400          EXP     1B18!400,0              ; 46
1077  001250' 000000  000000
1078  001251' 000000  401000          EXP     1B18!1000,0             ; 47
1079  001252' 000000  000000
1080  001253' 000000  402000          EXP     1B18!2000,0             ; 50
1081  001254' 000000  000000
1082  001255' 000000  404000          EXP     1B18!4000,0             ; 51
1083  001256' 000000  000000
1084  001257' 000000  410000          EXP     1B18!10000,0            ; 52
1085  001260' 000000  000000
1086  001261' 000000  420000          EXP     1B18!20000,0            ; 53
1087  001262' 000000  000000
```

```
 1088   001263' 000000  440000                  EXP     1B18!40000,0            ; 54
 1089   001264' 000000  000000
 1090   001265' 000000  500000                  EXP     1B18!100000,0           ; 55
 1091   001266' 000000  000000
 1092   001267' 000000  600000                  EXP     1B18!200000,0           ; 56
 1093   001270' 000000  000000
 1094   001271' 000000  400001                  EXP     1B18!400001,0           ; 57
 1095   001272' 000000  000000
 1096   001273' 000001  400000                  EXP     1B18!1000000,0          ; 60
 1097   001274' 000000  000000
 1098   001275' 000002  400000                  EXP     1B18!2000000,0          ; 61
 1099   001276' 000000  000000
 1100   001277' 000004  400000                  EXP     1B18!4000000,0          ; 62
 1101   001300' 000000  000000
 1102   001301' 000010  400000                  EXP     1B18!10000000,0         ; 63
 1103   001302' 000000  000000
 1104   001303' 000020  400000                  EXP     1B18!20000000,0         ; 64
 1105   001304' 000000  000000
 1106   001305' 000040  400000                  EXP     1B18!40000000,0         ; 65
 1107   001306' 000000  000000
 1108   001307' 000100  400000                  EXP     1B18!100000000,0        ; 66
 1109   001310' 000000  000000
 1110   001311' 000200  400000                  EXP     1B18!200000000,0        ; 67
 1111   001312' 000000  000000
 1112   001313' 000400  400000                  EXP     1B18!400000000,0        ; 70
 1113   001314' 000000  000000
 1114   001315' 001000  400000                  EXP     1B18!1000000000,0       ; 71
 1115   001316' 000000  000000
 1116   001317' 002000  400000                  EXP     1B18!2000000000,0       ; 72
 1117   001320' 000000  000000
 1118   001321' 004000  400000                  EXP     1B18!4000000000,0       ; 73
 1119   001322' 000000  000000
 1120   001323' 000000  000377                  EXP     377,377
 1121   001324' 000000  000377
```

```
                                          ;#*********************************************************************
                                          ;* TEST 10 - MBUS Error Test (MPROC module only)
                                          ;
                                          ; Description:  Cause drivers to be asserted on the MBUS one at a
                                          ;               time and verify that MBUS Error does  not  appear
                                          ;               in the CSR register, since only one  drive is  on
                                          ;               at once.
                                          ;
                                          ; Procedure:    Port Clear
                                          ;               Load microcode
                                          ;               Start up ucode at location 0
                                          ;               Stop ucode
                                          ;               Read CSR register
                                          ;               MBUS error set? yes - error
                                          ;                               no - ok
                                          ;
                                          ;               Do for starting locations 0,3,6,11
                                          ;
                                          ; Failure:      ---
                                          ;#*********************************************************************

                                          ; Test data

 001325' 254 00 0 00 001335'              TSTU10: JRST    TG10                    ; go start test
 001326' 210403  000010                           MPROC!MBUS!NDMP!ZMPROC!10       ; test mask
 001327' 001400' 011403'                          T10M,,[ASCIZ ^MBUS Error Test (MPROC module only)^]
 001330' 011413' 011325'                          [EXP MLAST!M3],,[EXP MLAST!E15]
 001331' 000000  001431'                          TSTU11                          ; failure test table
 001332' 000000  001507'                          TSTU12                          ;   ...
 001333' 000000  001643'                          TSTU13
 001334' 777777  777777                           -1

                                          ; Start test

 001335' 201 00 0 00 000000'              TG10:   MOVEI   Z8                      ; get address of module start
 001336' 260 17 0 00 001027*                      GO      TRACE                   ; handle trace output
 001337' 201 01 0 00 001400'                      MOVEI   1,T10M                  ; set up microcode address
 001340' 260 17 0 00 000000*                      GO      TLOAD                   ; load/verify it
 001341' 263 17 0 00 000000                       RTN                             ; failed - exit test

                                          ; Initialization

 001342' 400 15 0 00 000000               TL10:   SETZ    ERFLG,                  ; clear error flag
 001343' 260 17 0 00 001055*                      GO      IPACLR                  ; clear port
 001344' 400 06 0 00 000000                       SETZ    6,                      ; initialize table address pointer
```

```

                                        ; 1st segment of test (Segment A) - Turn on MBUS drivers one at a time

001345' 202 06 0 00 001054*             TA10:   MOVEM   6,TSTSUB                ; set up subtest number
001346' 350 00 0 00 001345*                     AOS     TSTSUB                  ;  ...
001347' 400 01 0 00 000000                      SETZ    1,                      ; timeout time (test is fast)
001350' 200 04 0 06 001372'                     MOVE    4,T10TAB(6)             ; get test descriptor
001351' 202 04 0 00 011165'                     MOVEM   4,SAVDAT                ; save it
001352' 554 04 0 00 000004                      HLRZ    4,4                     ; get starting address
001353' 260 17 0 00 000000*                     GO      AEXEC                   ; execute ALU type test
001354' 255 00 0 00 000000                      JFCL                            ; ignore any error
001355' 200 01 0 00 000000*                     MOVE    1,ALCSR                 ; get final CSR data
001356' 603 01 0 00 002000                      TLNE    1,(MBERR)               ; MBUS Error occurred?
001357' 474 15 0 00 000000                      SETO    ERFLG,                  ; yes - set error flag
001360' 607 01 0 00 004000                      TLNN    1,(CRAMPE)              ; CRAM PE occurred?
001361' 474 15 0 00 000000                      SETO    ERFLG,                  ; no - set error flag
001362' 027 00 0 00 001377'                     SCOPER  MA10                    ; print error message
001363' 254 00 0 00 001345'                     JRST    TA10                    ; loop on error
001364' 254 00 0 00 001371'                     JRST    TX10                    ; altmode exit
001365' 326 15 0 00 001371'                     JUMPN   ERFLG,TX10              ; error yet? yes - abort test

                                        ; Next driver

001366' 350 00 0 00 000006                      AOS     6                       ; point to next driver descriptor
001367' 332 00 0 06 001372'                     SKIPE   T10TAB(6)               ; done?
00137C' 254 00 0 00 001345'                     JRST    TA10                    ; no - loop till done

                                        ; End of test

001371' 263 17 0 00 000000              TX10:   RTN                             ; return

                                        ; Test table - Starting addr,,Error message addr

001372' 000000  011414'                 T10TAB: 0,,[ASCIZ /'OENA' signal asserted/]
001373' 000003  011421'                         3,,[ASCIZ /'Rd Local Mem' signal asserted/]
001374' 000006  011430'                         6,,[ASCIZ /'Ena EMUX' signal asserted by 'MPREADCSR'/]
001375' 000011  011441'                         11,,[ASCIZ /'Ena EMUX' signal asserted by 'MPREADEBUS'/]
001376' 000000  000000                          0

                                        ; Error message

001377' 270000  011104'                 MA10:   LAST!CALL!TXALL!MPMBU1          ; print test results
```

```
                                 ; Microcode:

001400' 000000  010000           T10M:   MWORD   <ADDR=0,JMAP,J=1>            ; 0
001401' 000000  000040
001402' 000100  002000                   MWORD   <CONT,OENA>                  ; 1 'OENA'
001403' 000000  000340
001404' 000200  020000                   MWORD   <JMAP,J=2,BAD>               ; 2
001405' 000000  000041
                                                                              ;
001406' 000300  040000                   MWORD   <JMAP,J=4>                   ; 3
001407' 000000  000040
001410' 000400  000000                   MWORD   <CONT,RDLM>                  ; 4 'RD LOCAL MEM'
001411' 000000  220340
001412' 000500  050000                   MWORD   <JMAP,J=5,BAD>               ; 5
001413' 000000  000041

001414' 000600  070000                   MWORD   <JMAP,J=7>                   ; 6
001415' 000000  000040
001416' 000700  000100                   MWORD   <CONT,SELE,MGC=100>          ; 7 'ENA EMUX' (by MPREADCSR)
001417' 000000  005340
001420' 001000  100000                   MWORD   <JMAP,J=10,BAD>              ; 10
001421' 000000  000041

001422' 001100  120000                   MWORD   <JMAP,J=12>                  ; 11
001423' 000000  000040
001424' 001200  000010                   MWORD   <CONT,SELE,MGC=10>           ; 12 'ENA EMUX' (by MPREADEBUS)
001425' 000000  005340
001426' 001300  130000                   MWORD   <JMAP,J=13,BAD>              ; 13
001427' 000000  000041
001430' 777777  777777                   -1
```

```
                                   ;#*********************************************************************
                                   ;* TEST 11 - MBUS Error Test (CBUS module only)
                                   ;
                                   ; Description:  Cause drivers to be asserted on the MBUS one at a
                                   ;               time and verify that MBUS Error does  not  appear
                                   ;               in the CSR register, since only one  drive is  on
                                   ;               at once.
                                   ;
                                   ; Procedure:    Port Clear
                                   ;               Load microcode
                                   ;               Start up ucode at location 0
                                   ;               Stop ucode
                                   ;               Read CSR register
                                   ;               MBUS error set? yes - error
                                   ;                               no - ok
                                   ;
                                   ;               Do for starting location 0
                                   ;
                                   ; Failure:      ---
                                   ;#*********************************************************************

                                   ; Test data

001431' 254 00 0 00 001440'        TSTU11: JRST    TG11                    ; go start test
001432' 210603  000011                     MPROC!MBUS!NDMP!NDCB!ZMPROC!11  ; test mask
001433' 001500' 011452'                    T11M,,[ASCIZ ^MBUS Error Test^]
001434' 011413' 011325'                    [EXP MLAST!M3],,[EXP MLAST!E15]
001435' 000000  001507'                    TSTU12                          ; failure test table
001436' 000000  001643'                    TSTU13                          ;   ...
001437' 777777  777777                     -1

                                   ; Start test

001440' 201 00 0 00 000000'        TG11:   MOVEI   Z8                      ; get address of module start
001441' 260 17 0 00 001336*                GO      TRACE                   ; handle trace output
001442' 201 01 0 00 001500'                MOVEI   1,T11M                  ; set up microcode address
001443' 260 17 0 00 001340*                GO      TLOAD                   ; load/verify it
001444' 263 17 0 00 000000                 RTN                             ; failed - exit test

                                   ; Initialization

001445' 400 15 0 00 000000         TL11:   SETZ    ERFLG,                  ; clear error flag
001446' 260 17 0 00 001343*                GO      IPACLR                  ; clear port
001447' 400 06 0 00 000000                 SETZ    6,                      ; initialize table address pointer
```

```

                                      ; 1st segment of test (Segment A) - Turn on MBUS drivers one at a time

001450' 202 06 0 00 001346*           TA11:   MOVEM   6,TSTSUB                ; set up subtest number
001451' 350 00 0 00 001450*                   AOS     TSTSUB                  ;   ...
001452' 400 01 0 00 000000                    SETZ    1,                      ; timeout time (test is fast)
001453' 200 04 0 06 001475'                   MOVE    4,T11TAB(6)             ; get test descriptor
001454' 202 04 0 00 011165'                   MOVEM   4,SAVDAT                ; save it
001455' 554 04 0 00 000004                    HLRZ    4,4                     ; get starting address
001456' 260 17 0 00 001353*                   GO      AEXEC                   ; execute ALU type test
001457' 255 00 0 00 000000                    JFCL                            ; ignore any error
001460' 200 01 0 00 001355*                   MOVE    1,ALCSR                 ; get final CSR data
001461' 603 01 0 00 002000                    TLNE    1,(MBERR)               ; MBUS Error occurred?
001462' 474 15 0 00 000000                    SETO    ERFLG,                  ; yes - set error flag
001463' 607 01 0 00 004000                    TLNN    1,(CRAMPE)              ; CRAM PE occurred?
001464' 474 15 0 00 000000                    SETO    ERFLG,                  ; no - set error flag
001465' 027 00 0 00 001477'                   SCOPER  MA11                    ; print error message
001466' 254 00 0 00 001450'                   JRST    TA11                    ; loop on error
001467' 254 00 0 00 001474'                   JRST    TX11                    ; altmode exit
001470' 326 15 0 00 001474'                   JUMPN   ERFLG,TX11              ; error yet? yes - abort test

                                      ; Next driver

001471' 350 00 0 00 000006                    AOS     6                       ; point to next driver descriptor
001472' 332 00 0 06 001475'                   SKIPE   T11TAB(6)               ; done?
001473' 254 00 0 00 001450'                   JRST    TA11                    ; no - loop till done

                                      ; End of test

001474' 263 17 0 00 000000            TX11:   RTN                             ; return

                                      ; Test table - Starting addr,,Error message addr

001475' 000000  011456'               T11TAB: 0,,[ASCIZ /'ENA CMUX' signal asserted by 'MPENACMUX'/]
001476' 000000  000000                        0

                                      ; Error message

001477' 270000  011104'               MA11:   LAST!CALL!TXALL!MPMBU1          ; print test results

                                      ; Microcode:

001500' 000000  010000                T11M:   MWORD   <ADDR=0,JMAP,J=1>               ; 0
001501' 000000  000040
001502' 000100  000200                        MWORD   <CONT,SELM,MGC=200>             ; 1 'Ena CMUX' (by MPENACMUX)
001503' 000000  002340
001504' 000200  020000                        MWORD   <JMAP,J=2,BAD>                  ; 2
001505' 000000  000041
001506' 777777  777777                        -1
```

```

                                  ;#******************************************************************
                                  ;* TEST 12 - MBUS Error Test (MPROC module only)
                                  ;
                                  ; Description:  Causes multiple drivers  to  be  asserted  on the
                                  ;               MBUS at the same  time and verify that MBUS Error
                                  ;               appears in the CSR register.
                                  ;
                                  ; Procedure:    Port Clear
                                  ;               Load microcode
                                  ;               Start up ucode at location 0
                                  ;               Stop ucode
                                  ;               Read CSR register
                                  ;               MBUS error set? no - error
                                  ;                               yes - ok
                                  ;
                                  ;               Do for starting locations 0,3,6,11,14,17,22,25
                                  ;
                                  ; Failure:      ---
                                  ;#******************************************************************

                                  ; Test data

001507' 254 00 0 00 001515'       TSTU12: JRST    TG12                    ; go start test
001510' 210403  000012                    MPROC!MBUS!NDMP!ZMPROC!12       ; test mask
001511' 001562' 011403'                   T12M,,[ASCIZ ^MBUS Error Test (MPROC module only)^]
001512' 011413' 011325'                   [EXP MLAST!M3],,[EXP MLAST!E15]
001513' 000000  001643'                   TSTU13                          ; failure test table
001514' 777777  777777                    -1                              ;   ...

                                  ; Start test

001515' 201 00 0 00 000000'       TG12:   MOVEI   Z8                      ; get address of module start
001516' 260 17 0 00 001441*               GO      TRACE                   ; handle trace output
001517' 201 01 0 00 001562'               MOVEI   1,T12M                  ; set up microcode address
001520' 260 17 0 00 001443*               GO      TLOAD                   ; load/verify it
001521' 263 17 0 00 000000                RTN                             ; failed - exit test

                                  ; Initialization

001522' 400 15 0 00 000000        TL12:   SETZ    ERFLG,                  ; clear error flag
001523' 260 17 0 00 001446*               GO      IPACLR                  ; clear port
001524' 400 06 0 00 000000                SETZ    6,                      ; initialize table address pointer
```

```

                                        ; 1st segment of test (Segment A) - Turn on multiple MBUS drivers

001525' 202 06 0 00 001451*     TA12:   MOVEM   6,TSTSUB                ; set up subtest number
001526' 350 00 0 00 001525*             AOS     TSTSUB                  ;   ...
001527' 400 01 0 00 000000              SETZ    1,                      ; timeout time (test is fast)
001530' 200 04 0 06 001550'             MOVE    4,T12TAB(6)             ; get test descriptor
001531' 202 04 0 00 011165'             MOVEM   4,SAVDAT                ; save it
001532' 554 04 0 00 000004              HLRZ    4,4                     ; get starting address
001533' 260 17 0 00 001456*             GO      AEXEC                   ; execute ALU type test
001534' 255 00 0 00 000000              JFCL                            ; ignore any error
001535' 200 01 0 00 001460*             MOVE    1,ALCSR                 ; get final CSR data
001536' 607 01 0 00 002000              TLNN    1,(MBERR)               ; MBUS Error occurred?
001537' 474 15 0 00 000000              SETO    ERFLG,                  ; no - set error flag
001540' 027 00 0 00 001561'             SCOPER  MA12                    ; print error message
001541' 254 00 0 00 001525'             JRST    TA12                    ; loop on error
001542' 254 00 0 00 001547'             JRST    TX12                    ; altmode exit
001543' 326 15 0 00 001547'             JUMPN   ERFLG,TX12              ; error yet? yes - abort test

                                        ; Next driver

001544' 350 00 0 00 000006              AOS     6                       ; point to next driver descriptor
001545' 332 00 0 06 001550'             SKIPE   T12TAB(6)               ; done?
001546' 254 00 0 00 001525'             JRST    TA12                    ; no - loop till done

                                        ; End of test

001547' 263 17 0 00 000000      TX12:   RTN                             ; return

                                        ; Test table - Starting addr,,Error message addr

001550' 000000  011467'         T12TAB: 0,,[ASCIZ /'OENA','RDLM' asserted/]
001551' 000003  011474'                 3,,[ASCIZ /'OENA','ENA EMUX' (by MPREADCSR) asserted/]
001552' 000006  011505'                 6,,[ASCIZ /'OENA','ENA EMUX' (by MPREADEBUS) asserted/]
001553' 000011  011516'                 11,,[ASCIZ /'RDLM','ENA EMUX' (by MPREADCSR) asserted/]
001554' 000014  011527'                 14,,[ASCIZ /'RDLM','ENA EMUX' (by MPLOADEBUS) asserted/]
001555' 000017  011540'                 17,,[ASCIZ /'OENA','RDLM','ENA EMUX' (by MPREADCSR) asserted/]
001556' 000022  011552'                 22,,[ASCIZ /'OENA','RDLM','ENA EMUX' (by MPLOADEBUS) asserted/]
001557' 000025  011564'                 25,,[ASCIZ /'OENA','RDLM','ENA EMUX' (by MPLOADEBUS,MPREADEBUS) asserted/]
001560' 000000  000000                  0
                                ; Error message

001561' 270000  011133'         MA12:   LAST!CALL!TXALL!MPMBU2          ; print test results

                                ; Microcode:

001562' 000000  010000          T12M:   MWORD   <ADDR=0,JMAP,J=1>               ; 0
001563' 000000  000040
001564' 000100  002000                  MWORD   <CONT,OENA,RDLM>                ; 1 'OENA','RDLM'
001565' 000000  220340
001566' 000200  020000                  MWORD   <JMAP,J=2,BAD>                  ; 2
001567' 000000  000041

001570' 000300  040000                  MWORD   <JMAP,J=4>                      ; 3
001571' 000000  000040
```

```
001572' 000400  002100          MWORD   <CONT,OENA,SELE,MGC=100>        ; 4 'OENA','ENA EMUX'
001573' 000000  005340
001574' 000500  050000          MWORD   <JMAP,J=5,BAD>                  ; 5
001575' 000000  000041

001576' 000600  070000          MWORD   <JMAP,J=7>                      ; 6
001577' 000000  000040
001600' 000700  002010          MWORD   <CONT,OENA,SELE,MGC=10>         ; 7 'OENA','ENA EMUX'
001601' 000000  005340
001602' 001000  100000          MWORD   <JMAP,J=10,BAD>                 ; 10
001603' 000000  000041

001604' 001100  120000          MWORD   <JMAP,J=12>                     ; 11
001605' 000000  000040
001606' 001200  000100          MWORD   <CONT,RDLM,SELE,MGC=100>        ; 12 'RDLM','ENA EMUX'
001607' 000000  225340
001610' 001300  130000          MWORD   <JMAP,J=13,BAD>                 ; 13
001611' 000000  000041

001612' 001400  150000          MWORD   <JMAP,J=15>                     ; 14
001613' 000000  000040
001614' 001500  000010          MWORD   <CONT,RDLM,SELE,MGC=10>         ; 15 'RDLM','ENA EMUX'
001615' 000000  225340
001616' 001600  160000          MWORD   <JMAP,J=16,BAD>                 ; 16
001617' 000000  000041

001620' 001700  200000          MWORD   <JMAP,J=20>                     ; 17
001621' 000000  000040
001622' 002000  002100          MWORD   <CONT,OENA,RDLM,SELE,MGC=100>   ; 20 'OENA','RDLM','ENA EMUX'
001623' 000000  225340
001624' 002100  210000          MWORD   <JMAP,J=21,BAD>                 ; 21
001625' 000000  000041

001626' 002200  230000          MWORD   <JMAP,J=23>                     ; 22
001627' 000000  000040
001630' 002300  002010          MWORD   <CONT,OENA,RDLM,SELE,MGC=10>    ; 23 'OENA','RDLM','ENA EMUX'
001631' 000000  225340
001632' 002400  240000          MWORD   <JMAP,J=24,BAD>                 ; 24
001633' 000000  000041

001634' 002500  260000          MWORD   <JMAP,J=26>                     ; 25
001635' 000000  000040
001636' 002600  002110          MWORD   <CONT,OENA,RDLM,SELE,MGC=110>   ; 26 'OENA','RDLM','ENA EMUX'
001637' 000000  225340
001640' 002700  270000          MWORD   <JMAP,J=27,BAD>                 ; 27
001641' 000000  000041
001642' 777777  777777          -1
```

```
                                       ;#*************************************************************************
                                       ;* TEST 13 - MBUS Error Test (CBUS module only)
                                       ;
                                       ; Description:  Causes multiple drivers  to  be  asserted  on the
                                       ;               MBUS at the same  time and verify that MBUS Error
                                       ;               appears in the CSR register.
                                       ;
                                       ; Procedure:    Port Clear
                                       ;               Load microcode
                                       ;               Start up ucode at location 0
                                       ;               Stop ucode
                                       ;               Read CSR register
                                       ;               MBUS error set? yes - error
                                       ;                               no - ok
                                       ;
                                       ;               Do for starting locations 0,3,6,11,14,17,22,25,30,33
                                       ;
                                       ; Failure:      ---
                                       ;#*************************************************************************

                                       ; Test data

001643' 254 00 0 00 001650'            TSTU13: JRST    TG13                            ; go start test
001644' 210603  000013                         MPROC!MBUS!NDMP!NDCB!ZMPROC!13  ; test mask
001645' 001710' 011452'                        T13M,,[ASCIZ ^MBUS Error Test^]
001646' 011413' 011325'                        [EXP MLAST!M3],,[EXP MLAST!E15]
001647' 777777  777777                         -1                              ; failure test table

                                       ; Start test

001650' 201 00 0 00 000000'            TG13:   MOVEI   Z8                              ; get address of module start
001651' 260 17 0 00 001516*                    GO      TRACE                           ; handle trace output
001652' 201 01 0 00 001710'                    MOVEI   1,T13M                          ; set up microcode address
001653' 260 17 0 00 001520*                    GO      TLOAD                           ; load/verify it
001654' 263 17 0 00 000000                     RTN                                     ; failed - exit test

                                       ; Initialization

001655' 400 15 0 00 000000             TL13:   SETZ    ERFLG,                          ; clear error flag
001656' 260 17 0 00 001523*                    GO      IPACLR                          ; clear port
001657' 400 06 0 00 000000                     SETZ    6,                              ; initialize table address pointer

                                       ; 1st segment of test (Segment A) - Turn on MBUS drivers one at a time

001660' 202 06 0 00 001526*            TA13:   MOVEM   6,TSTSUB                        ; set up subtest number
001661' 350 00 0 00 001660*                    AOS     TSTSUB                          ;   ...
001662' 400 01 0 00 000000                     SETZ    1,                              ; timeout time (test is fast)
001663' 200 04 0 06 001703'                    MOVE    4,T13TAB(6)                     ; get test descriptor
001664' 202 04 0 00 011165'                    MOVEM   4,SAVDAT                        ; save it
001665' 554 04 0 00 000004                     HLRZ    4,4                             ; get starting address
001666' 260 17 0 00 001533*                    GO      AEXEC                           ; execute ALU type test
001667' 255 00 0 00 000000                     JFCL                                    ; ignore any error
```

```
001670' 200 01 0 00 001535*             MOVE    1,ALCSR                 ; get final CSR data
001671' 607 01 0 00 002000              TLNN    1,(MBERR)               ; MBUS Error occurred?
001672' 474 15 0 00 000000              SETO    ERFLG,                  ; no - set error flag
001673' 027 00 0 00 001707'             SCOPER  MA13                    ; print error message
001674' 254 00 0 00 001660'             JRST    TA13                    ; loop on error
001675' 254 00 0 00 001702'             JRST    TX13                    ; altmode exit
001676' 326 15 0 00 001702'             JUMPN   ERFLG,TX13              ; error yet? yes - abort test


                                ; Next driver

001677' 350 00 0 00 000006              AOS     6                       ; point to next driver descriptor
001700' 332 00 0 06 001703'             SKIPE   T13TAB(6)               ; done?
001701' 254 00 0 00 001660'             JRST    TA13                    ; no - loop till done


                                ; End of test

001702' 263 17 0 00 000000      TX13:   RTN                             ; return

                                ; Test table - Starting addr,,Error message addr

001703' 000000  011601'         T13TAB: 0,,[ASCIZ /'OENA','ENA CMUX (Ena Cmux)' asserted/]
001704' 000003  011611'                 3,,[ASCIZ /'RDLM','ENA CMUX (Ena Cmux)' asserted/]
001705' 000006  011621'                 6,,[ASCIZ /'OENA','RDLM','ENA CMUX (Ena Cmux)' asserted/]
001706' 000000  000000                  0

                                ; Error message

001707' 270000  011133'         MA13:   LAST!CALL!TALL!MPMBU2           ; print test results

                                ; Microcode:

001710' 000000  010000          T13M:   MWORD   <ADDR=0,JMAP,J=1>               ; 0
001711' 000000  000040
001712' 000100  002200                  MWORD   <CONT,OENA,SELM,MGC=200>        ; 1 OENA,EnaCMUX
001713' 000000  002340
001714' 000200  020000                  MWORD   <JMAP,J=2,BAD>                  ; 2
001715' 000000  000041

001716' 000300  040000                  MWORD   <JMAP,J=4>                      ; 3
001717' 000000  000040
001720' 000400  000200                  MWORD   <CONT,RDLM,SELM,MGC=200>        ; 4 RDLM,EnaCMUX
001721' 000000  222340
001722' 000500  050000                  MWORD   <JMAP,J=5,BAD>                  ; 5
001723' 000000  000041

001724' 000600  070000                  MWORD   <JMAP,J=7>                      ; 6
001725' 000000  000040
001726' 000700  002200                  MWORD   <CONT,OENA,RDLM,SELM,MGC=200>   ; 7 OENA,RDLM,EnaCMUX
001727' 000000  222340
001730' 001000  100000                  MWORD   <JMAP,J=10,BAD>                 ; 10
001731' 000000  000041
001732' 777777  777777                  -1
```

```
                                        ;#***********************************************************************
                                        ;* TEST 14 - CCCbusAvail Test
                                        ;
                                        ; Description:  Verify that condition code 'CCCbusAvail'  can  be
                                        ;               tested  by  the  microsequencer  for  conditional
                                        ;               branching, and that the condition code can be set
                                        ;               and reset by the hardware conditions which should
                                        ;               do so.
                                        ;
                                        ; Procedure:    Port Clear
                                        ;               Start up port
                                        ;               UC> perform an operation and then verify that
                                        ;                   CCCbusAvail is not asserted or asserted
                                        ;               UC> if passed - set bit 12 (DISABLE Complete)
                                        ;               UC> if failed - set bit 13 (ENABLE Complete)
                                        ;               Read CSR and determine test disposition
                                        ;
                                        ;               Repeat with several test cases:
                                        ;                 CC is cleared initially
                                        ;                 CC is set by a read in progress
                                        ;                 CC is set by a write in progress
                                        ;                 CC is cleared by completion of a read
                                        ;                 CC is cleared by completion of a write
                                        ;                 CC is cleared by 'Clrcccode'
                                        ;
                                        ; Failure:      ---
                                        ;#***********************************************************************

                                        ; Test data

001733' 254 00 0 00 001745'             TSTU14: JRST    TG14                    ; go start test
001734' 200403  000014                          MPROC!NDMP!ZMPROC!14            ; test mask
001735' 002060' 011632'                         T14M,,[ASCIZ ^CCCbusAvail Test^]
001736' 011636' 011640'                         [EXP M6,MLAST!C5],,[EXP M7,C15,C7,C8,MLAST!C1]
001737' 000000000000#                           TSTC3                           ; failure test table
001740' 000000000000#                           TSTC4                           ;   ...
001741' 000000  005512'                         TSTU30
001742' 000000  006307'                         TSTU32
001743' 000000  006603'                         TSTU33
001744' 777777  777777                          -1

                                        ; Start test

001745' 201 00 0 00 000000'             TG14:   MOVEI   Z8                      ; get address of module start
001746' 260 17 0 00 001651*                     GO      TRACE                   ; handle trace output
001747' 201 01 0 00 002060'                     MOVEI   1,T14M                  ; set up microcode address
001750' 260 17 0 00 001653*                     GO      TLOAD                   ; load/verify it
001751' 263 17 0 00 000000                      RTN                             ; failed - exit test
```

```


                                        ; Initialization

001752' 400 15 0 00 000000      TL14:   SETZ    ERFLG,                  ; clear error flag
001753' 201 06 0 00 001770'             MOVEI   6,TS14                  ; get test table address
001754' 402 00 0 00 001661*             SETZM   TSTSUB                  ; initialize subtest number
001755' 476 00 0 00 000000*             SETOM   TADDR                   ; clear start address

                                        ; Loop on test execute table entries

001756' 260 17 0 00 001656*     TA14:   GO      IPACLR                  ; clear port
001757' 260 17 0 00 000000*     TB14:   GO      TEXEC                   ; execute table entry
001760' 254 00 0 00 001767'              JRST   TX14                    ; end of sstep table
001761' 254 00 0 00 001757'              JRST   TB14                    ; keep looping after call
001762' 474 15 0 00 000000               SETO   ERFLG,                  ; error occurred

                                        ; Handle error printouts and scope looping

001763' 027 00 0 00 002044'             SCOPER  MA14                    ; print error message
001764' 254 00 0 00 001752'             JRST    TL14                    ; loop on error
001765' 254 00 0 00 001767'             JRST    TX14                    ; altmode exit
001766' 322 15 0 00 001756'             JUMPE   ERFLG,TA14              ; do next sstep table entry

                                        ; End of test

001767' 263 17 0 00 000000      TX14:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

                                        ; CC is cleared initially

001770' 140000  002021'         TS14:   TTABLE  (TCALL,TS14IN)          ; initialize port
001771' 040000  000000                  TTABLE  (TSTART,0)              ; start up port
001772' 240002  000000                  TTABLE  (TCHECK,2)              ; check if completed
001773' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; CC is set by a read in progress

001774' 140000  002031'                 TTABLE  (TCALL,TS14RW)          ; start up a read
001775' 040000  000010                  TTABLE  (TSTART,10)             ; start up port
001776' 240002  000000                  TTABLE  (TCHECK,2)              ; check if completed
001777' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; CC is set by a write in progress

002000' 140000  002031'                 TTABLE  (TCALL,TS14RW)          ; start up a write
002001' 040000  000020                  TTABLE  (TSTART,20)             ; start up port
002002' 240002  000000                  TTABLE  (TCHECK,2)              ; check if completed
002003' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

```

```
                                        ; CC is cleared by completion of a read

002004' 140000  002031'                         TTABLE  (TCALL,TS14RW)          ; start up a read
002005' 040000  000030                          TTABLE  (TSTART,30)             ; start up port
002006' 240002  000000                          TTABLE  (TCHECK,2)              ; check if completed
002007' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; CC is cleared by completion of a write

002010' 140000  002031'                         TTABLE  (TCALL,TS14RW)          ; start up a write
002011' 040000  000040                          TTABLE  (TSTART,40)             ; start up port
002012' 240002  000000                          TTABLE  (TCHECK,2)              ; check if completed
002013' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; CC is cleared by 'Clrcccode'

002014' 140000  002031'                         TTABLE  (TCALL,TS14RW)          ; start up a write
002015' 040000  000050                          TTABLE  (TSTART,50)             ; start up port
002016' 240002  000000                          TTABLE  (TCHECK,2)              ; check if completed
002017' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet
002020' 000000  000000                          TTABLE  (TLAST)

002021' 260 17 0 00 001756*             TS14IN: GO      IPACLR                  ; clear port
002022' 201 01 0 00 005000                      MOVEI   1,5000                  ; get starting address
002023' 242 01 0 00 000001                      LSH     1,1                     ; position properly
002024' 260 17 0 00 000452*                     GO      LDRAR                   ; set up starting address
002025' 201 01 0 00 000010                      MOVEI   1,MPRUN                 ; get 'MPRUN' bit
002026' 260 17 0 00 000457*                     GO      LDCSR                   ; start up the port
002027' 260 17 0 00 002021*                     GO      IPACLR                  ; stop the port
002030' 263 17 0 00 000000                      RTN                             ; return - done

002031' 201 14 0 00 000001              TS14RW: MOVEI   PAT,1                   ; get data pattern
002032' 201 01 0 00 000002                      MOVEI   1,2                     ; get transfer type
002033' 201 02 0 00 000010                      MOVEI   2,10                    ; get transfer length
002034' 260 17 0 00 000000*                     GO      BUFGEN                  ; generate buffer contents
002035' 200 01 0 00 011645'                     MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
002036' 260 17 0 00 000000*                     GO      CHINIT                  ; initialize software
002037' 551 01 0 00 000000*                     HRRZI   1,BUFF                  ; buffer address
002040' 201 02 0 00 000010                      MOVEI   2,10                    ; word count
002041' 400 03 0 00 000000                      SETZ    3,                      ; clear zero fill specifier
002042' 260 17 0 00 000000*                     GO      GENCCW                  ; generate a CCW list
002043' 263 17 0 00 000000                      RTN                             ; return
```

```
                                      ; Error messages

002044' 140000  011646'               MA14:   MSG!TXNOT![ASCIZ /Condition Code 'CC CBUS Avail' not /]
002045' 240000  002047'                       CALL!TXNOT!MA14LP                ; print subtest description
002046' 000000000000#                         LAST!CALL!TXALL!TTPNT            ; print test data

002047' 200 01 0 00 001754*           MA14LP: MOVE    1,TSTSUB                 ; get subtest number
002050' 256 00 0 01 002051'                   XCT     MA14L-1(1)               ; print message
002051' 263 17 0 00 000000                    RTN                              ; return

002052' 037 00 0 00 011656'           MA14L:  TMSG    <cleared initially>
002053' 037 00 0 00 011662'                   TMSG    <set by a read in progress>
002054' 037 00 0 00 011670'                   TMSG    <set by a write in progress>
002055' 037 00 0 00 011676'                   TMSG    <cleared by completion of a read>
002056' 037 00 0 00 011705'                   TMSG    <cleared by completion of a write>
002057' 037 00 0 00 011714'                   TMSG    <cleared by 'Clrcccode'>

                                      ; Microcode:

                                      ;       CC is cleared initially

002060' 000000  010000                T14M:   MWORD   <ADDR=0,JMAP,J=1>                        ; 0
002061' 000000  000040
002062' 000152  000000                        MWORD   <CJP,J=5200,CENA,CCCA>                   ; 1
002063' 000400  000060
002064' 000251  000000                        MWORD   <JMAP,J=5100>                            ; 2
002065' 000000  000040

                                      ;       CC is set by a read in progress

002066' 001000  110020                        MWORD   <ADDR=10,JMAP,J=11,SELM,MGC=20>          ; 10
002067' 000000  002040
002070' 001100  000200                        MWORD   <CONT,D=1,SELC,MGC=200>                  ; 11
002071' 001000  004340
002072' 001251  000000                        MWORD   <CJP,J=5100,D=1,CENA,CCCA>               ; 12
002073' 001400  000060
002074' 001300  120000                        MWORD   <JMAP,J=12,D=1>                          ; 13
002075' 001000  000040

                                      ;       CC is set by a write in progress

002076' 002000  210020                        MWORD   <ADDR=20,JMAP,J=21,SELM,MGC=20>          ; 20
002077' 000000  002040
002100' 002100  000220                        MWORD   <CONT,D=1,SELC,MGC=220>                  ; 21
002101' 001000  004340
002102' 002251  000000                        MWORD   <CJP,J=5100,D=1,CENA,CCCA>               ; 22
002103' 001400  000060
002104' 002300  220000                        MWORD   <JMAP,J=22,D=1>                          ; 23
002105' 001000  000040

                                      ;       CC is cleared after completion of a read

002106' 003000  310020                        MWORD   <ADDR=30,JMAP,J=31,SELM.MGC=20>          ; 30
002107' 000000  002040
```

```
002110' 003100  000200          MWORD   <CONT,D=1,SELC,MGC=200>                 ; 31
002111' 001000  004340
002112' 003200  340000          MWORD   <CJP,J=34,D=1,CENA,CCCA>                ; 32
002113' 001400  000060
002114' 003300  320000          MWORD   <JMAP,J=32,D=1>                         ; 33
002115' 001000  000040
002116' 003400  000004          MWORD   <CONT,D=1,SELC,MGC=4>                   ; 34
002117' 001000  004340
002120' 003552  000000          MWORD   <CJP,J=5200,CENA,CCCA>                  ; 35
002121' 000400  000060
002122' 003651  000000          MWORD   <JMAP,J=5100>                           ; 36
002123' 000000  000040

                        ;       CC is cleared by completion of a write

002124' 004000  410020          MWORD   <ADDR=40,JMAP,J=41,SELM,MGC=20>         ; 40
002125' 000000  002040
002126' 004100  000220          MWORD   <CONT,D=1,SELC,MGC=220>                 ; 41
002127' 001000  004340
002130' 004200  440000          MWORD   <CJP,J=44,D=1,CENA,CCCA>                ; 42
002131' 001400  000060
002132' 004300  420000          MWORD   <JMAP,J=42,D=1>                         ; 43
002133' 001000  000040
002134' 004400  000004          MWORD   <CONT,D=1,SELC,MGC=4>                   ; 44
002135' 001000  004340
002136' 004552  000000          MWORD   <CJP,J=5200,CENA,CCCA>                  ; 45
002137' 000400  000060
002140' 004651  000000          MWORD   <JMAP,J=5100>                           ; 46
002141' 000000  000040

                        ;       CC is cleared by 'Clrcccode'

002142' 005000  510020          MWORD   <ADDR=50,JMAP,J=51,SELM,MGC=20>         ; 50
002143' 000000  002040
002144' 005100  000220          MWORD   <CONT,D=1,SELC,MGC=220>                 ; 51
002145' 001000  004340
002146' 005200  540000          MWORD   <CJP,J=54,D=1,CENA,CCCA>                ; 52
002147' 001400  000060
002150' 005300  520000          MWORD   <JMAP,J=52,D=1>                         ; 53
002151' 001000  000040
002152' 005400  000020          MWORD   <CONT,D=1,SELM,MGC=20>                  ; 54
002153' 001000  002340
002154' 005552  000000          MWORD   <CJP,J=5200,CENA,CCCA>                  ; 55
002155' 000400  000060
002156' 005651  000000          MWORD   <JMAP,J=5100>                           ; 56
002157' 000000  000040

                        ; Initialization

002160' 500050  010100          MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>    ; 5000
002161' 000000  004040
002162' 500150  010000          MWORD   <JMAP,J=5001>                           ; 5001
002163' 000000  000040

                        ; Subroutine - Passed - Write 1 to bit 12 of CSR register
```

```
1837
1838  002164' 510000  000000                    MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>           ; 5100
1839  002165' 742001  000340
1840  002166' 510100  000000                    MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                ; 5101
1841  002167' 302001  000740
1842  002170' 510200  260000                    MWORD   <LDCT,J=26>                                ; 5102
1843  002171' 000000  000300
1844  002172' 510351  030000                    MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>         ; 5103
1845  002173' 102021  000220
1846  002174' 510400  002004                    MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>      ; 5104
1847  002175' 431020  005340
1848  002176' 510551  070040                    MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>     ; 5105
1849  002177' 001400  015060
1850  002200' 510651  050000                    MWORD   <JMAP,J=5105,D=1>                          ; 5106
1851  002201' 001000  000040
1852  002202' 510700  000200                    MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>         ; 5107
1853  002203' 431010  005340
1854  002204' 511051  100000                    MWORD   <JMAP,J=5110,D=1>                          ; 5110
1855  002205' 001000  000040
1856
1857                                    ; Subroutine - Failed - Write 1 to bit 13 of CSR register
1858
1859  002206' 520000  000000                    MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>           ; 5200
1860  002207' 742001  000340
1861  002210' 520100  000000                    MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                ; 5201
1862  002211' 302001  000740
1863  002212' 520200  250000                    MWORD   <LDCT,J=25>                                ; 5202
1864  002213' 000000  000300
1865  002214' 520352  030000                    MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>         ; 5203
1866  002215' 102021  000220
1867  002216' 520400  002004                    MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>      ; 5204
1868  002217' 431020  005340
1869  002220' 520552  070040                    MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>     ; 5205
1870  002221' 001400  015060
1871  002222' 520652  050000                    MWORD   <JMAP,J=5205,D=1>                          ; 5206
1872  002223' 001000  000040
1873  002224' 520700  000200                    MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>         ; 5207
1874  002225' 431010  005340
1875  002226' 521052  100000                    MWORD   <JMAP,J=5210,D=1>                          ; 5210
1876  002227' 001000  000040
1877  002230' 777777  777777                    -1
```

```
                                                    ;#*********************************************************************
                                                    ;* TEST 15 - CCGrntCSR Test
                                                    ;
                                                    ; Description:   Verify that condition  code  'CCGrntCSR'  can  be
                                                    ;                tested  by  the  microsequencer  for  conditional
                                                    ;                branching, and that the condition code can be set
                                                    ;                and reset by the hardware conditions which should
                                                    ;                do so.
                                                    ;
                                                    ; Procedure:     Port Clear
                                                    ;                Start up port
                                                    ;                UC> perform an operation and then verify that
                                                    ;                       CCGrntCSR is not asserted or asserted
                                                    ;                UC> if passed - set bit 12 (DISABLE Complete)
                                                    ;                UC> if failed - set bit 13 (ENABLE Complete)
                                                    ;                Read CSR and determine test disposition
                                                    ;
                                                    ;                Repeat with several test cases:
                                                    ;                  CC is cleared initially
                                                    ;                  CC is set after MP requests access to CSR
                                                    ;                  CC is cleared after MP reads CSR
                                                    ;                  CC is cleared after MP writes CSR
                                                    ;                  CC is still set after KL attempts to read the CSR
                                                    ;                  CC is still set after KL attempts to write the CSR
                                                    ;
                                                    ; Failure:       ---
                                                    ;#*********************************************************************

                                                    ; Test data

002231' 254 00 0 00 002243'                         TSTU15: JRST    TG15                            ; go start test
002232' 200403  000015                                      MPROC!NDMP!ZMPROC!15                    ; test mask
002233' 002334' 011721'                                     T15M,,[ASCIZ ^CCGrntCSR Test^]
002234' 011724' 011726'                                     [EXP M6,MLAST!E3],,[EXP M7,MLAST!E7]
002235' 000000  002467'                                     TSTU16                                  ; failure test table
002236' 000000  003315'                                     TSTU17                                  ;   ...
002237' 000000  003525'                                     TSTU20
002240' 000000  003677'                                     TSTU21
002241' 000000  004070'                                     TSTU22
002242' 777777  777777                                      -1

                                                    ; Start test

002243' 201 00 0 00 000000'                         TG15:   MOVEI   Z8                              ; get address of module start
002244' 260 17 0 00 001746*                                 GO      TRACE                           ; handle trace output
002245' 201 01 0 00 002334'                                 MOVEI   1,T15M                          ; set up microcode address
002246' 260 17 0 00 001750*                                 GO      TLOAD                           ; load/verify it
002247' 263 17 0 00 000000                                  RTN                                     ; failed - exit test
```

```

                                        ; Initialization

002250' 400 15 0 00 000000      TL15:   SETZ    ERFLG,                  ; clear error flag
002251' 201 06 0 00 002266'             MOVEI   6,TS15                  ; get test table address
002252' 402 00 0 00 002047*             SETZM   TSTSUB                  ; initialize subtest number
002253' 476 00 0 00 001755*             SETOM   TADDR                   ; clear start address

                                        ; Loop on test execute table entries

002254' 260 17 0 00 002027*     TA15:   GO      IPACLR                  ; clear port
002255' 260 17 0 00 001757*     TB15:   GO      TEXEC                   ; execute table entry
002256' 254 00 0 00 002265'              JRST   TX15                    ; end of sstep table
002257' 254 00 0 00 002255'              JRST   TB15                    ; keep looping after call
002260' 474 15 0 00 000000               SETO   ERFLG,                  ; error occurred

                                        ; Handle error printouts and scope looping

002261' 027 00 0 00 002320'             SCOPER  MA15                    ; print error message
002262' 254 00 0 00 002250'             JRST    TL15                    ; loop on error
002263' 254 00 0 00 002265'             JRST    TX15                    ; altmode exit
002264' 322 15 0 00 002254'             JUMPE   ERFLG,TA15              ; do next sstep table entry

                                        ; End of test

002265' 263 17 0 00 000000      TX15:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

                                        ; CC is cleared initially

002266' 040000  000000          TS15:   TTABLE  (TSTART,0)              ; start up port
002267' 240002  000000                  TTABLE  (TCHECK,2)              ; check if completed
002270' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; CC is set after MP requests access to CSR

002271' 040000  000010                  TTABLE  (TSTART,10)             ; start up port
002272' 240002  000000                  TTABLE  (TCHECK,2)              ; check if completed
002273' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; CC is cleared after MP reads CSR

002274' 040000  000020                  TTABLE  (TSTART,20)             ; start up port
002275' 240002  000000                  TTABLE  (TCHECK,2)              ; check if completed
002276' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

```

```

                                    ; CC is cleared after MP writes CSR

002277' 040000  000030                      TTABLE  (TSTART,30)             ; start up port
002300' 240002  000000                      TTABLE  (TCHECK,2)              ; check if completed
002301' 340000  000000                      TTABLE  (TEXIT)                 ; exit if error yet

                                    ; CC is still set after KL does a DATAI

002302' 040000  000040                      TTABLE  (TSTART,40)             ; start up port
002303' 140000  002313'                     TTABLE  (TCALL,TDATAI)          ; read CSR
002304' 240002  000000                      TTABLE  (TCHECK,2)              ; check if completed
002305' 340000  000000                      TTABLE  (TEXIT)                 ; exit if error yet

                                    ; CC is still set after KL does a DATAO

002306' 040000  000040                      TTABLE  (TSTART,40)             ; start up port
002307' 140000  002315'                     TTABLE  (TCALL,TDATAO)          ; read CSR
002310' 240002  000000                      TTABLE  (TCHECK,2)              ; check if completed
002311' 340000  000000                      TTABLE  (TEXIT)                 ; exit if error yet
002312' 000000  000000                      TTABLE  (TLAST)                 ; end of table

002313' 260 17 0 00 000460*         TDATAI: GO      RDLAR                   ; read LAR
002314' 263 17 0 00 000000                  RTN                             ; return

002315' 400 01 0 00 000000          TDATAO: SETZ    1,                      ; clear data
002316' 260 17 0 00 002024*                 GO      LDRAR                   ; write RAR
002317' 263 17 0 00 000000                  RTN                             ; return

                                    ; Error messages

002320' 140000  011730'             MA15:   MSG!TXNOT![ASCIZ /Condition Code 'CC Grnt CSR' not /]
002321' 240000  002323'                     CALL!TXNOT!MA15LP               ; print subtest description
002322' 000000000000#                       LAST!CALL!TXALL!TTPNT           ; print test data

002323' 200 01 0 00 002252*         MA15LP: MOVE    1,TSTSUB                ; get subtest number
002324' 256 00 0 01 002325'                 XCT     MA15L-1(1)              ; print message
002325' 263 17 0 00 000000                  RTN                             ; return

002326' 037 00 0 00 011656'         MA15L:  TMSG    <cleared initially>
002327' 037 00 0 00 011737'                 TMSG    <set after MP requests access to CSR>
002330' 037 00 0 00 011747'                 TMSG    <cleared after MP reads CSR>
002331' 037 00 0 00 011755'                 TMSG    <cleared after MP writes CSR>
                                            TMSG    <still set after KL attempts to read
002332' 037 00 0 00 011763'           the CSR>
                                            TMSG    <still set after KL attempts to write
002333' 037 00 0 00 011775'           the CSR>
```

```
                                    ; Microcode:

                                    ;       CC is cleared initially

002334' 000000  010000              T15M:   MWORD   <ADDR=0,JMAP,J=1>                       ; 0
002335' 000000  000040
002336' 000152  000000                      MWORD   <CJP,J=5200,CENA,CCGC>                  ; 1
002337' 000400  010060
002340' 000251  000000                      MWORD   <JMAP,J=5100>                           ; 2
002341' 000000  000040

                                    ;       CC is set after MP requests access to CSR

002342' 001000  110040                      MWORD   <ADDR=10,JMAP,SELE,MGC=40,J=11>         ; 10
002343' 000000  005040
002344' 001151  000040                      MWORD   <CJP,J=5100,CENA,CCGC,MGC=40,SELE>
002345' 000400  015060
002346' 001200  110000                      MWORD   <JMAP,J=11>                             ; 12
002347' 000000  000040

                                    ;       CC is cleared after MP reads CSR

002350' 002000  210040                      MWORD   <ADDR=20,JMAP,SELE,MGC=40,J=21>         ; 20
002351' 000000  005040
002352' 002100  230000                      MWORD   <CJP,J=23,CENA,CCGC>                    ; 21
002353' 000400  010060
002354' 002200  210000                      MWORD   <JMAP,J=21>                             ; 22
002355' 000000  000040
002356' 002300  000100                      MWORD   <CONT,SELE,MGC=100>                     ; 23
002357' 000000  005340
002360' 002452  000000                      MWORD   <CJP,J=5200,CENA,CCGC>                  ; 24
002361' 000400  010060
002362' 002551  000000                      MWORD   <JMAP,J=5100>                           ; 25
002363' 000000  000040

                                    ;       CC is cleared after MP writes CSR

002364' 003000  310040                      MWORD   <ADDR=30,JMAP,SELE,MGC=40,J=31>         ; 30
002365' 000000  005040
002366' 003100  330000                      MWORD   <CJP,J=33,CENA,CCGC>                    ; 31
002367' 000400  010060
002370' 003200  310000                      MWORD   <JMAP,J=31>                             ; 32
002371' 000000  000040
002372' 003300  000004                      MWORD   <CONT,SOA,AND,D=1,SELE,MGC=4>           ; 33
002373' 441000  005340
002374' 003400  000200                      MWORD   <CONT,SELE,MGC=200>                     ; 34
002375' 000000  005340
002376' 003552  000000                      MWORD   <CJP,J=5200,CENA,CCGC>                  ; 35
002377' 000400  010060
002400' 003651  000000                      MWORD   <JMAP,J=5100>                           ; 36
002401' 000000  000040

                                    ;       CC is still set after KL attempts to read the CSR
                                    ;       CC is still set after KL attempts to write the CSR
```

```

 002402' 004000  410040          MWORD   <ADDR=40,JMAP,SELE,MGC=40,J=41>            ; 40
 002403' 000000  005040
 002404' 004100  430000          MWORD   <CJP,J=43,CENA,CCER>                       ; 41
 002405' 000400  100060
 002406' 004200  410000          MWORD   <JMAP,J=41>                                ; 42
 002407' 000000  000040
 002410' 004300  460000          MWORD   <CJP,J=46,CENA,CCGC>                       ; 43
 002411' 000400  010060
 002412' 004400  000200          MWORD   <CONT,SELE,MGC=200>                        ; 44
 002413' 000000  005340
 002414' 004552  000000          MWORD   <JMAP,J=5200>                              ; 45
 002415' 000000  000040
 002416' 004600  000200          MWORD   <CONT,SELE,MGC=200>                        ; 46
 002417' 000000  005340
 002420' 004751  000000          MWORD   <JMAP,J=5100>                              ; 47
 002421' 000000  000040

                         ; Subroutine - Passed - Write 1 to bit 12 of CSR register

 002422' 510000  000000          MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>           ; 5100
 002423' 742001  000340
 002424' 510100  000000          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                ; 5101
 002425' 302001  000740
 002426' 510200  260000          MWORD   <LDCT,J=26>                                ; 5102
 002427' 000000  000300
 002430' 510351  030000          MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>         ; 5103
 002431' 102021  000220
 002432' 510400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>      ; 5104
 002433' 431020  005340
 002434' 510551  070040          MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>     ; 5105
 002435' 001400  015060
 002436' 510651  050000          MWORD   <JMAP,J=5105,D=1>                          ; 5106
 002437' 001000  000040
 002440' 510700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>         ; 5107
 002441' 431010  005340
 002442' 511051  100000          MWORD   <JMAP,J=5110,D=1>                          ; 5110
 002443' 001000  000040

                         ; Subroutine   Failed - Write 1 to bit 13 of CSR register

 002444' 520000  000000          MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>           ; 5200
 002445' 742001  000340
 002446' 520100  000000          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                ; 5201
 002447' 302001  000740
 002450' 520200  250000          MWORD   <LDCT,J=25>                                ; 5202
 002451' 000000  000300
 002452' 520352  030000          MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>         ; 5203
 002453' 102021  000220
 002454' 520400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>      ; 5204
 002455' 431020  005340
 002456' 520552  070040          MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>     ; 5205
 002457' 001400  015060
 002460' 520652  050000          MWORD   <JMAP,J=5205,D=1>                          ; 5206
 002461' 001000  000040
```

```
002462' 520700  000200                  MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5207
002463' 431010  005340
002464' 521052  100000                  MWORD   <JMAP,J=5210,D=1>                       ; 5210
002465' 001000  000040
002466' 777777  777777                  -1
```

```
                                  ;#****************************************************************
                                  ;* TEST 16 - CCFEQ0 Test
                                  ;
                                  ; Description:  Verify that condition code 'CCFEQ0' can be tested
                                  ;               by  the  microsequencer  for  use  in conditional
                                  ;               branching, and that the condition code can be set
                                  ;               and reset by the hardware conditions which should
                                  ;               do so.
                                  ;
                                  ; Procedure:    Port Clear
                                  ;               Start up port
                                  ;               UC> gate -1's out of the 2901's and verify that
                                  ;                   the CCFEQ0 is not asserted or asserted
                                  ;               UC> if passed - set bit 12 (DISABLE Complete)
                                  ;               UC> if failed - set bit 13 (ENABLE Complete)
                                  ;               Read CSR and determine test disposition
                                  ;
                                  ;               Repeat with several test cases:
                                  ;                 -1's
                                  ;                 floating 1's
                                  ;                 0's
                                  ;
                                  ; Failure:      ---
                                  ;#****************************************************************

                                  ; Test data

002467' 254 00 0 00 002501'       TSTU16: JRST    TG16                    ; go start test
002470' 200403  000016                    MPROC!NDMP!ZMPROC!16            ; test mask
002471' 002560' 012007'                   T16M,,[ASCIZ ^CCFEQ0 Test^]
002472' 012012' 012014'                   [EXP M6,MLAST!E23],,[EXP MLAST!M7]
002473' 000000  003315'                   TSTU17                          ; failure test table
002474' 000000  003525'                   TSTU20                          ;   ...
002475' 000000  003677'                   TSTU21
002476' 000000  004070'                   TSTU22
002477' 000000  004261'                   TSTU23
002500' 777777  777777                    -1

                                  ; Start test

002501' 201 00 0 00 000000'       TG16:   MOVEI   Z8                      ; get address of module start
002502' 260 17 0 00 002244*               GO      TRACE                   ; handle trace output
002503' 201 01 0 00 002560'               MOVEI   1,T16M                  ; set up microcode address
002504' 260 17 0 00 002246*               GO      TLOAD                   ; load/verify it
002505' 263 17 0 00 000000                RTN                             ; failed - exit test
```

```
                                     ; Initialization

 002506' 400 15 0 00 000000          TL16:   SETZ    ERFLG,                  ; clear error flag
 002507' 201 06 0 00 002524'                  MOVEI   6,TS16                  ; get test table address
 002510' 402 00 0 00 002323*                  SETZM   TSTSUB                  ; initialize subtest number
 002511' 476 00 0 00 002253*                  SETOM   TADDR                   ; clear start address

                                     ; Loop on test execute table entries

 002512' 260 17 0 00 002254*          TA16:   GO      IPACLR                  ; clear port
 002513' 260 17 0 00 002255*          TB16:   GO      TEXEC                   ; execute table entry
 002514' 254 00 0 00 002523'                   JRST   TX16                    ; end of sstep table
 002515' 254 00 0 00 002513'                   JRST   TB16                    ; keep looping after call
 002516' 474 15 0 00 000000                    SETO   ERFLG,                  ; error occurred

                                     ; Handle error printouts and scope looping

 002517' 027 00 0 00 002546'                  SCOPER  MA16                    ; print error message
 002520' 254 00 0 00 002506'                  JRST    TL16                    ; loop on error
 002521' 254 00 0 00 002523'                  JRST    TX16                    ; altmode exit
 002522' 322 15 0 00 002512'                  JUMPE   ERFLG,TA16              ; do next sstep table entry

                                     ; End of test

 002523' 263 17 0 00 000000          TX16:    RTN                             ; return

                                     ; Test Execute Table, as:  (CMD,parameters)

 002524' 200000  002531'              TS16:   TTABLE  (TCALLC,TS16AD)         ; set up starting address
 002525' 040000  077777                       TTABLE  (TSTART,77777)          ; start up port
 002526' 240002  000000                       TTABLE  (TCHECK,2)              ; check if completed
 002527' 340000  000000                       TTABLE  (TEXIT)                 ; exit if error yet
 002530' 300000  002524'                      TTABLE  (TJRST,TS16)            ; loop till done

 002531' 350 01 0 00 002511*          TS16AD: AOS     1,TADDR                 ; increment address
 002532' 307 01 0 00 000045                   CAIG    1,45                    ; done yet?
 002533' 350 00 0 17 000000                   AOS     (P)                     ; no - set up return+2
 002534' 476 00 0 00 011165'                  SETOM   SAVDAT#                 ; set data to 1's
 002535' 322 01 0 00 002545'                  JUMPE   1,TS16AX                ; zero - yes - exit
 002536' 402 00 0 00 011165'                  SETZM   SAVDAT                  ; set data to 0's
 002537' 306 01 0 00 000045                   CAIN    1,45                    ; zero's?
 002540' 263 17 0 00 000000                   RTN                             ; yes - return
 002541' 370 00 0 00 000001                   SOS     1                       ; calculate number of shifts
 002542' 201 00 0 00 000001                   MOVEI   1                       ; get a 1 bit
 002543' 242 00 0 01 000000                   LSH     (1)                     ; shift (for floating 1)
 002544' 202 00 0 00 011165'                  MOVEM   SAVDAT                  ; save it
 002545' 263 17 0 00 000000          TS16AX:  RTN                             ; return
```

```
                                          ; Error messages

002546' 140000  012015'                   MA16:   MSG!TXNOT![ASCIZ /Condition code 'CC FEQLO'/]
002547' 240000  002551'                           CALL!TXNOT!MA16PN              ; print data used
002550' 000000000000#                             LAST!CALL!TXALL!TTPNT          ; print test data

002551' 336 00 0 00 011165'               MA16PN: SKIPN   SAVDAT
                                                  JRST    [TMSG   < should be set (data 0's) but isn't.>
                                                           RTN]
002552' 254 00 0 00 012033'                       TMSG    < should not be set (data >
002553' 037 00 0 00 012035'                       MOVE    SAVDAT
002554' 200 00 0 00 011165'                       PNTHW
002555' 037 13 0 00 000000                        TMSG    <) but is.>
002556' 037 00 0 00 012043'                       RTN
002557' 263 17 0 00 000000

                                          ; Microcode:

002560' 000001  000000                    T16M:   MWORD   <ADDR=0,JMAP,J=100>              ; 0 - 1's
002561' 000000  000040
002562' 000102  000000                            MWORD   <JMAP,J=200>                     ; 1 - 1
002563' 000000  000040
002564' 000203  000000                            MWORD   <JMAP,J=300>                     ; 2 - 2
002565' 000000  000040
002566' 000304  000000                            MWORD   <JMAP,J=400>                     ; 3 - 4
002567' 000000  000040
002570' 000405  000000                            MWORD   <JMAP,J=500>                     ; 4 - 10
002571' 000000  000040
002572' 000506  000000                            MWORD   <JMAP,J=600>                     ; 5 - 20
002573' 000000  000040
002574' 000607  000000                            MWORD   <JMAP,J=700>                     ; 6 - 40
002575' 000000  000040
002576' 000710  000000                            MWORD   <JMAP,J=1000>                    ; 7 - 100
002577' 000000  000040
002600' 001011  000000                            MWORD   <JMAP,J=1100>                    ; 10 - 200
002601' 000000  000040
002602' 001112  000000                            MWORD   <JMAP,J=1200>                    ; 11 - 400
002603' 000000  000040
002604' 001213  000000                            MWORD   <JMAP,J=1300>                    ; 12 - 1000
002605' 000000  000040
002606' 001314  000000                            MWORD   <JMAP,J=1400>                    ; 13 - 2000
002607' 000000  000040
002610' 001415  000000                            MWORD   <JMAP,J=1500>                    ; 14 - 4000
002611' 000000  000040
002612' 001516  000000                            MWORD   <JMAP,J=1600>                    ; 15 - 10000
002613' 000000  000040
002614' 001617  000000                            MWORD   <JMAP,J=1700>                    ; 16 - 20000
002615' 000000  000040
002616' 001720  000000                            MWORD   <JMAP,J=2000>                    ; 17 - 40000
002617' 000000  000040
002620' 002021  000000                            MWORD   <JMAP,J=2100>                    ; 20 - 100000
002621' 000000  000040
002622' 002122  000000                            MWORD   <JMAP,J=2200>                    ; 21 - 200000
002623' 000000  000040
002624' 002223  000000                            MWORD   <JMAP,J=2300>                    ; 22 - 400000
```

```
002625' 000000  000040
002626' 002324  000000          MWORD   <JMAP,J=2400>                           ; 23 - 1,,0
002627' 000000  000040
002630' 002425  000000          MWORD   <JMAP,J=2500>                           ; 24 - 2,,0
002631' 000000  000040
002632' 002526  000000          MWORD   <JMAP,J=2600>                           ; 25 - 4,,0
002633' 000000  000040
002634' 002627  000000          MWORD   <JMAP,J=2700>                           ; 26 - 10,,0
002635' 000000  000040
002636' 002730  000000          MWORD   <JMAP,J=3000>                           ; 27 - 20,,0
002637' 000000  000040
002640' 003031  000000          MWORD   <JMAP,J=3100>                           ; 30 - 40,,0
002641' 000000  000040
002642' 003132  000000          MWORD   <JMAP,J=3200>                           ; 31 - 100,,0
002643' 000000  000040
002644' 003233  000000          MWORD   <JMAP,J=3300>                           ; 32 - 200,,0
002645' 000000  000040
002646' 003334  000000          MWORD   <JMAP,J=3400>                           ; 33 - 400,,0
002647' 000000  000040
002650' 003435  000000          MWORD   <JMAP,J=3500>                           ; 34 - 1000,,0
002651' 000000  000040
002652' 003536  000000          MWORD   <JMAP,J=3600>                           ; 35 - 2000,,0
002653' 000000  000040
002654' 003637  000000          MWORD   <JMAP,J=3700>                           ; 36 - 4000,,0
002655' 000000  000040
002656' 003740  000000          MWORD   <JMAP,J=4000>                           ; 37 - 10000,,0
002657' 000000  000040
002660' 004041  000000          MWORD   <JMAP,J=4100>                           ; 40 - 20000,,0
002661' 000000  000040
002662' 004142  000000          MWORD   <JMAP,J=4200>                           ; 41 - 40000,,0
002663' 000000  000040
002664' 004243  000000          MWORD   <JMAP,J=4300>                           ; 42 - 100000,,0
002665' 000000  000040
002666' 004344  000000          MWORD   <JMAP,J=4400>                           ; 43 - 200000,,0
002667' 000000  000040
002670' 004445  000000          MWORD   <JMAP,J=4500>                           ; 44 - 400000,,0
002671' 000000  000040
002672' 004546  000000          MWORD   <JMAP,J=4600>                           ; 45 - 0's
002673' 000000  000040

                        ; Data pattern 1's

002674' 010000  000000          MWORD   <ADDR=100,CONT,SDO,AND,D=2>             ; 100
002675' 742000  000340
002676' 010152  002000          MWORD   <CJP,J=5200,SAB,XNOR,D=1,CENA,OENA,CCFZ>; 101
002677' 171400  020060
002700' 010251  000000          MWORD   <JMAP,J=5100>                           ; 102
002701' 000000  000040

                        ; Data pattern floating 1's - 1

002702' 020000  000000          MWORD   <ADDR=200,CONT,SDO,AND,D=2>             ; 200
002703' 742000  000340
002704' 020100  000000          MWORD   <CONT,SAB,PLUS,CRY,D=2>                 ; 201
002705' 102000  000740
```

```
002706' 020252  002000          MWORD   <CJP,J=5200,SOA,OR,D=1,CENA,OENA,CCFZ>  ; 202
002707' 431400  020060
002710' 020351  000000          MWORD   <JMAP,J=5100>                           ; 203
002711' 000000  000040

                        ; Data pattern floating 1's - 2

002712' 030000  000000          MWORD   <ADDR=300,CONT,SDO,AND,D=2>             ; 300
002713' 742000  000340
002714' 030100  000000          MWORD   <LDCT,J=0,SAB,PLUS,CRY,D=2>             ; 301
002715' 102000  000700
002716' 030253  000000          MWORD   <JMAP,J=5300>                           ; 302
002717' 000000  000040

                        ; Data pattern floating 1's - 4

002720' 040000  000000          MWORD   <ADDR=400,CONT,SDO,AND,D=2>             ; 400
002721' 742000  000340
002722' 040100  010000          MWORD   <LDCT,J=1,SAB,PLUS,CRY,D=2>             ; 401
002723' 102000  000700
002724' 040253  000000          MWORD   <JMAP,J=5300>                           ; 402
002725' 000000  000040

                        ; Data pattern floating 1's - 10

002726' 050000  000000          MWORD   <ADDR=500,CONT,SDO,AND,D=2>             ; 500
002727' 742000  000340
002730' 050100  020000          MWORD   <LDCT,J=2,SAB,PLUS,CRY,D=2>             ; 501
002731' 102000  000700
002732' 050253  000000          MWORD   <JMAP,J=5300>                           ; 502
002733' 000000  000040

                        ; Data pattern floating 1's - 20

002734' 060000  000000          MWORD   <ADDR=600,CONT,SDO,AND,D=2>             ; 600
002735' 742000  000340
002736' 060100  030000          MWORD   <LDCT,J=3,SAB,PLUS,CRY,D=2>             ; 601
002737' 102000  000700
002740' 060253  000000          MWORD   <JMAP,J=5300>                           ; 602
002741' 000000  000040

                        ; Data pattern floating 1's - 40

002742' 070000  000000          MWORD   <ADDR=700,CONT,SDO,AND,D=2>             ; 700
002743' 742000  000340
002744' 070100  040000          MWORD   <LDCT,J=4,SAB,PLUS,CRY,D=2>             ; 701
002745' 102000  000700
002746' 070253  000000          MWORD   <JMAP,J=5300>                           ; 702
002747' 000000  000040

                        ; Data pattern floating 1's - 100

002750' 100000  000000          MWORD   <ADDR=1000,CONT,SDO,AND,D=2>            ; 1000
002751' 742000  000340
002752' 100100  050000          MWORD   <LDCT,J=5,SAB,PLUS,CRY,D=2>             ; 1001
```

```
2395  002753' 102000  000700
2396  002754' 100253  000000                    MWORD   <JMAP,J=5300>                           ; 1002
2397  002755' 000000  000040
2398
2399                                    ; Data pattern floating 1's - 200
2400
2401  002756' 110000  000000                    MWORD   <ADDR=1100,CONT,SDO,AND,D=2>            ; 1100
2402  002757' 742000  000340
2403  002760' 110100  060000                    MWORD   <LDCT,J=6,SAB,PLUS,CRY,D=2>             ; 1101
2404  002761' 102000  000700
2405  002762' 110253  000000                    MWORD   <JMAP,J=5300>                           ; 1102
2406  002763' 000000  000040
2407
2408                                    ; Data pattern floating 1's - 400
2409
2410  002764' 120000  000000                    MWORD   <ADDR=1200,CONT,SDO,AND,D=2>            ; 1200
2411  002765' 742000  000340
2412  002766' 120100  070000                    MWORD   <LDCT,J=7,SAB,PLUS,CRY,D=2>             ; 1201
2413  002767' 102000  000700
2414  002770' 120253  000000                    MWORD   <JMAP,J=5300>                           ; 1202
2415  002771' 000000  000040
2416
2417                                    ; Data pattern floating 1's - 1000
2418
2419  002772' 130000  000000                    MWORD   <ADDR=1300,CONT,SDO,AND,D=2>            ; 1300
2420  002773' 742000  000340
2421  002774' 130100  100000                    MWORD   <LDCT,J=10,SAB,PLUS,CRY,D=2>            ; 1301
2422  002775' 102000  000700
2423  002776' 130253  000000                    MWORD   <JMAP,J=5300>                           ; 1302
2424  002777' 000000  000040
2425
2426                                    ; Data pattern floating 1's - 2000
2427
2428  003000' 140000  000000                    MWORD   <ADDR=1400,CONT,SDO,AND,D=2>            ; 1400
2429  003001' 742000  000340
2430  003002' 140100  110000                    MWORD   <LDCT,J=11,SAB,PLUS,CRY,D=2>            ; 1401
2431  003003' 102000  000700
2432  003004' 140253  000000                    MWORD   <JMAP,J=5300>                           ; 1402
2433  003005' 000000  000040
2434
2435                                    ; Data pattern floating 1's - 4000
2436
2437  003006' 150000  000000                    MWORD   <ADDR=1500,CONT,SDO,AND,D=2>            ; 1500
2438  003007' 742000  000340
2439  003010' 150100  120000                    MWORD   <LDCT,J=12,SAB,PLUS,CRY,D=2>            ; 1501
2440  003011' 102000  000700
2441  003012' 150253  000000                    MWORD   <JMAP,J=5300>                           ; 1502
2442  003013' 000000  000040
2443
2444                                    ; Data pattern floating 1's - 10000
2445
2446  003014' 160000  000000                    MWORD   <ADDR=1600,CONT,SDO,AND,D=2>            ; 1600
2447  003015' 742000  000340
2448  003016' 160100  130000                    MWORD   <LDCT,J=13,SAB,PLUS,CRY,D=2>            ; 1601
2449  003017' 102000  000700
```

```
003020' 160253  000000                  MWORD   <JMAP,J=5300>                           ; 1602
003021' 000000  000040


                                ; Data pattern floating 1's - 20000

003022' 170000  000000                  MWORD   <ADDR=1700,CONT,SDO,AND,D=2>            ; 1700
003023' 742000  000340
003024' 170100  140000                  MWORD   <LDCT,J=14,SAB,PLUS,CRY,D=2>            ; 1701
003025' 102000  000700
003026' 170253  000000                  MWORD   <JMAP,J=5300>                           ; 1702
003027' 000000  000040


                                ; Data pattern floating 1's - 40000

003030' 200000  000000                  MWORD   <ADDR=2000,CONT,SDO,AND,D=2>            ; 2000
003031' 742000  000340
003032' 200100  150000                  MWORD   <LDCT,J=15,SAB,PLUS,CRY,D=2>            ; 2001
003033' 102000  000700
003034' 200253  000000                  MWORD   <JMAP,J=5300>                           ; 2002
003035' 000000  000040


                                ; Data pattern floating 1's - 100000

003036' 210000  000000                  MWORD   <ADDR=2100,CONT,SDO,AND,D=2>            ; 2100
003037' 742000  000340
003040' 210100  160000                  MWORD   <LDCT,J=16,SAB,PLUS,CRY,D=2>            ; 2101
003041' 102000  000700
003042' 210253  000000                  MWORD   <JMAP,J=5300>                           ; 2102
003043' 000000  000040


                                ; Data pattern floating 1's - 200000

003044' 220000  000000                  MWORD   <ADDR=2200,CONT,SDO,AND,D=2>            ; 2200
003045' 742000  000340
003046' 220100  170000                  MWORD   <LDCT,J=17,SAB,PLUS,CRY,D=2>            ; 2201
003047' 102000  000700
003050' 220253  000000                  MWORD   <JMAP,J=5300>                           ; 2202
003051' 000000  000040


                                ; Data pattern floating 1's - 400000

003052' 230000  000000                  MWORD   <ADDR=2300,CONT,SDO,AND,D=2>            ; 2300
003053' 742000  000340
003054' 230100  200000                  MWORD   <LDCT,J=20,SAB,PLUS,CRY,D=2>            ; 2301
003055' 102000  000700
003056' 230253  000000                  MWORD   <JMAP,J=5300>                           ; 2302
003057' 000000  000040


                                ; Data pattern floating 1's - 1,,000000

003060' 240000  000000                  MWORD   <ADDR=2400,CONT,SDO,AND,D=2>            ; 2400
003061' 742000  000340
003062' 240100  210000                  MWORD   <LDCT,J=21,SAB,PLUS,CRY,D=2>            ; 2401
003063' 102000  000700
003064' 240253  000000                  MWORD   <JMAP,J=5300>                           ; 2402
```

```
003065' 000000  000040

                                   ; Data pattern floating 1's - 2,,000000

003066' 250000  000000                  MWORD   <ADDR=2500,CONT,SDO,AND,D=2>             ; 2500
003067' 742000  000340
003070' 250100  220000                  MWORD   <LDCT,J=22,SAB,PLUS,CRY,D=2>             ; 2501
003071' 102000  000700
003072' 250253  000000                  MWORD   <JMAP,J=5300>                            ; 2502
003073' 000000  000040

                                   ; Data pattern floating 1's - 4,,000000

003074' 260000  000000                  MWORD   <ADDR=2600,CONT,SDO,AND,D=2>             ; 2600
003075' 742000  000340
003076' 260100  230000                  MWORD   <LDCT,J=23,SAB,PLUS,CRY,D=2>             ; 2601
003077' 102000  000700
003100' 260253  000000                  MWORD   <JMAP,J=5300>                            ; 2602
003101' 000000  000040

                                   ; Data pattern floating 1's - 10,,000000

003102' 270000  000000                  MWORD   <ADDR=2700,CONT,SDO,AND,D=2>             ; 2700
003103' 742000  000340
003104' 270100  240000                  MWORD   <LDCT,J=24,SAB,PLUS,CRY,D=2>             ; 2701
003105' 102000  000700
003106' 270253  000000                  MWORD   <JMAP,J=5300>                            ; 2702
003107' 000000  000040

                                   ; Data pattern floating 1's - 20,,000000

003110' 300000  000000                  MWORD   <ADDR=3000,CONT,SDO,AND,D=2>             ; 3000
003111' 742000  000340
003112' 300100  250000                  MWORD   <LDCT,J=25,SAB,PLUS,CRY,D=2>             ; 3001
003113' 102000  000700
003114' 300253  000000                  MWORD   <JMAP,J=5300>                            ; 3002
003115' 000000  000040

                                   ; Data pattern floating 1's - 40,,000000

003116' 310000  000000                  MWORD   <ADDR=3100,CONT,SDO,AND,D=2>             ; 3100
003117' 742000  000340
003120' 310100  260000                  MWORD   <LDCT,J=26,SAB,PLUS,CRY,D=2>             ; 3101
003121' 102000  000700
003122' 310253  000000                  MWORD   <JMAP,J=5300>                            ; 3102
003123' 000000  000040

                                   ; Data pattern floating 1's - 100,,000000

003124' 320000  000000                  MWORD   <ADDR=3200,CONT,SDO,AND,D=2>             ; 3200
003125' 742000  000340
003126' 320100  270000                  MWORD   <LDCT,J=27,SAB,PLUS,CRY,D=2>             ; 3201
003127' 102000  000700
003130' 320253  000000                  MWORD   <JMAP,J=5300>                            ; 3202
003131' 000000  000040
```

```

                                ; Data pattern floating 1's - 200,,000000

003132' 330000  000000                  MWORD   <ADDR=3300,CONT,SDO,AND,D=2>            ; 3300
003133' 742000  000340
003134' 330100  300000                  MWORD   <LDCT,J=30,SAB,PLUS,CRY,D=2>            ; 3301
003135' 102000  000700
003136' 330253  000000                  MWORD   <JMAP,J=5300>                           ; 3302
003137' 000000  000040

                                ; Data pattern floating 1's - 400,,000000

003140' 340000  000000                  MWORD   <ADDR=3400,CONT,SDO,AND,D=2>            ; 3400
003141' 742000  000340
003142' 340100  310000                  MWORD   <LDCT,J=31,SAB,PLUS,CRY,D=2>            ; 3401
003143' 102000  000700
003144' 340253  000000                  MWORD   <JMAP,J=5300>                           ; 3402
003145' 000000  000040

                                ; Data pattern floating 1's - 1000,,000000

003146' 350000  000000                  MWORD   <ADDR=3500,CONT,SDO,AND,D=2>            ; 3500
003147' 742000  000340
003150' 350100  320000                  MWORD   <LDCT,J=32,SAB,PLUS,CRY,D=2>            ; 3501
003151' 102000  000700
003152' 350253  000000                  MWORD   <JMAP,J=5300>                           ; 3502
003153' 000000  000040

                                ; Data pattern floating 1's - 2000,,000000

003154' 360000  000000                  MWORD   <ADDR=3600,CONT,SDO,AND,D=2>            ; 3600
003155' 742000  000340
003156' 360100  330000                  MWORD   <LDCT,J=33,SAB,PLUS,CRY,D=2>            ; 3601
003157' 102000  000700
003160' 360253  000000                  MWORD   <JMAP,J=5300>                           ; 3602
003161' 000000  000040

                                ; Data pattern floating 1's - 4000,,000000

003162' 370000  000000                  MWORD   <ADDR=3700,CONT,SDO,AND,D=2>            ; 3700
003163' 742000  000340
003164' 370100  340000                  MWORD   <LDCT,J=34,SAB,PLUS,CRY,D=2>            ; 3701
003165' 102000  000700
003166' 370253  000000                  MWORD   <JMAP,J=5300>                           ; 3702
003167' 000000  000040

                                ; Data pattern floating 1's - 10000,,000000

003170' 400000  000000                  MWORD   <ADDR=4000,CONT,SDO,AND,D=2>            ; 4000
003171' 742000  000340
003172' 400100  350000                  MWORD   <LDCT,J=35,SAB,PLUS,CRY,D=2>            ; 4001
003173' 102000  000700
003174' 400253  000000                  MWORD   <JMAP,J=5300>                           ; 4002
003175' 000000  000040

```

```
                                          ; Data pattern floating 1's - 20000,,000000

003176' 410000  000000                          MWORD   <ADDR=4100,CONT,SDO,AND,D=2>             ; 4100
003177' 742000  000340
003200' 410100  360000                          MWORD   <LDCT,J=36,SAB,PLUS,CRY,D=2>             ; 4101
003201' 102000  000700
003202' 410253  000000                          MWORD   <JMAP,J=5300>                            ; 4102
003203' 000000  000040

                                          ; Data pattern floating 1's - 40000,,000000

003204' 420000  000000                          MWORD   <ADDR=4200,CONT,SDO,AND,D=2>             ; 4200
003205' 742000  000340
003206' 420100  370000                          MWORD   <LDCT,J=37,SAB,PLUS,CRY,D=2>             ; 4201
003207' 102000  000700
003210' 420253  000000                          MWORD   <JMAP,J=5300>                            ; 4202
003211' 000000  000040

                                          ; Data pattern floating 1's - 100000,,000000

003212' 430000  000000                          MWORD   <ADDR=4300,CONT,SDO,AND,D=2>             ; 4300
003213' 742000  000340
003214' 430100  400000                          MWORD   <LDCT,J=40,SAB,PLUS,CRY,D=2>             ; 4301
003215' 102000  000700
003216' 430253  000000                          MWORD   <JMAP,J=5300>                            ; 4302
003217' 000000  000040

                                          ; Data pattern floating 1's - 200000,,000000

003220' 440000  000000                          MWORD   <ADDR=4400,CONT,SDO,AND,D=2>             ; 4400
003221' 742000  000340
003222' 440100  410000                          MWORD   <LDCT,J=41,SAB,PLUS,CRY,D=2>             ; 4401
003223' 102000  000700
003224' 440253  000000                          MWORD   <JMAP,J=5300>                            ; 4402
003225' 000000  000040

                                          ; Data pattern floating 1's - 400000,,000000

003226' 450000  000000                          MWORD   <ADDR=4500,CONT,SDO,AND,D=2>             ; 4500
003227' 742000  000340
003230' 450100  420000                          MWORD   <LDCT,J=42,SAB,PLUS,CRY,D=2>             ; 4501
003231' 102000  000700
003232' 450245  020000                          MWORD   <RPCT,J=4502,SOA,OR,D=7>                 ; 4502
003233' 437000  000220
003234' 450353  010000                          MWORD   <JMAP,J=5301>                            ; 4503
003235' 000000  000040

                                          ; Data pattern 0's

003236' 460051  000000                          MWORD   <ADDR=4600,CJP,J=5100,SDO,AND,D=1,CENA,CCFZ>
003237' 741400  020060
003240' 460152  000000                          MWORD   <JMAP,J=5200>                            ; 4601
003241' 000000  000040

                                          ; Common code for floating 1's
```

```

 003242' 530053  000000                 MWORD   <ADDR=5300,RPCT,J=5300,SAB,PLUS,D=2>      ; 5300
 003243' 102000  000220
 003244' 530152  002000                 MWORD   <CJP,J=5200,SOA,OR,D=1,CENA,OENA,CCFZ>   ; 5301
 003245' 431400  020060
 003246' 530251  000000                 MWORD   <JMAP,J=5100>                             ; 5302
 003247' 000000  000040

                                ; Subroutine - Passed - Write 1 to bit 12 of CSR register

 003250' 510000  000000                 MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>          ; 5100
 003251' 742001  000340
 003252' 510100  000000                 MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>               ; 5101
 003253' 302001  000740
 003254' 510200  260000                 MWORD   <LDCT,J=26>                               ; 5102
 003255' 000000  000300
 003256' 510351  030000                 MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>        ; 5103
 003257' 102021  000220
 003260' 510400  002004                 MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>     ; 5104
 003261' 431020  005340
 003262' 510551  070040                 MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>   ; 5105
 003263' 001400  015060
 003264' 510651  050000                 MWORD   <JMAP,J=5105,D=1>                         ; 5106
 003265' 001000  000040
 003266' 510700  000200                 MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>        ; 5107
 003267' 431010  005340
 003270' 511051  100000                 MWORD   <JMAP,J=5110,D=1>                         ; 5110
 003271' 001000  000040

                                ; Subroutine - Failed - Write 1 to bit 13 of CSR register

 003272' 520000  000000                 MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>          ; 5200
 003273' 742001  000340
 003274' 520100  000000                 MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>               ; 5201
 003275' 302001  000740
 003276' 520200  250000                 MWORD   <LDCT,J=25>                               ; 5202
 003277' 000000  000300
 003300' 520352  030000                 MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>        ; 5203
 003301' 102021  000220
 003302' 520400  002004                 MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>     ; 5204
 003303' 431020  005340
 003304' 520552  070040                 MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>   ; 5205
 003305' 001400  015060
 003306' 520652  050000                 MWORD   <JMAP,J=5205,D=1>                         ; 5206
 003307' 001000  000040
 003310' 520700  000200                 MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>        ; 5207
 003311' 431010  005340
 003312' 521052  100000                 MWORD   <JMAP,J=5210,D=1>                         ; 5210
 003313' 001000  000040
 003314' 777777  777777                 -1
```

```
                                        ;#**********************************************************************
                                        ;* TEST 17 - CCCSRChng Test
                                        ;
                                        ; Description:  Verify that condition  code  'CCCSRChng'  can  be
                                        ;               tested  by  the  microsequencer  for  conditional
                                        ;               branching, and that the condition code can be set
                                        ;               and reset by the hardware conditions which should
                                        ;               do so.
                                        ;
                                        ; Procedure:    Port Clear
                                        ;               Start up port
                                        ;               UC> perform an operation and then verify that
                                        ;                   CCCSRChng is not asserted or asserted
                                        ;               UC> if passed - set bit 12 (DISABLE Complete)
                                        ;               UC> if failed - set bit 13 (ENABLE Complete)
                                        ;               Read CSR and determine test disposition
                                        ;
                                        ;               Repeat with several test cases:
                                        ;                 CC is set initially (KL last wrote CSR)
                                        ;                 CC is not set after port reads CSR
                                        ;                 CC is set after an EBUS PE
                                        ;                 CC stays set if port writes CSR
                                        ;
                                        ; Failure:      ---
                                        ;#**********************************************************************

                                        ; Test data

003315' 254 00 0 00 003327'             TSTU17: JRST    TG17                    ; go start test
003316' 200403  000017                          MPROC!NDMP!ZMPROC!17            ; test mask
003317' 003404' 012045'                         T17M,,[ASCIZ ^CCCSRChng Test^]
003320' 012050' 012052'                         [EXP M6,MLAST!E8],,[EXP M7,E1,E6,MLAST!E16]
003321' 000000  003525'                         TSTU20                          ; failure test table
003322' 000000  003677'                         TSTU21                          ;   ...
003323' 000000  004070'                         TSTU22
003324' 000000  004261'                         TSTU23
003325' 000000  004452'                         TSTU24
003326' 777777  777777                          -1

                                        ; Start test

003327' 201 00 0 00 000000'             TG17:   MOVEI   Z8                      ; get address of module start
003330' 260 17 0 00 002502*                     GO      TRACE                   ; handle trace output
003331' 201 01 0 00 003404'                     MOVEI   1,T17M                  ; set up microcode address
003332' 260 17 0 00 002504*                     GO      TLOAD                   ; load/verify it
003333' 263 17 0 00 000000                      RTN                             ; failed - exit test
```

```

                                        ; Initialization

003334' 400 15 0 00 000000      TL17:   SETZ    ERFLG,                  ; clear error flag
003335' 201 06 0 00 003352'             MOVEI   6,TS17                  ; get test table address
003336' 402 00 0 00 002510*             SETZM   TSTSUB                  ; initialize subtest number
003337' 476 00 0 00 002531*             SETOM   TADDR                   ; clear start address

                                        ; Loop on test execute table entries

003340' 260 17 0 00 002512*     TA17:   GO      IPACLR                  ; clear port
003341' 260 17 0 00 002513*     TB17:   GO      TEXEC                   ; execute table entry
003342' 254 00 0 00 003351'              JRST   TX17                    ; end of sstep table
003343' 254 00 0 00 003341'              JRST   TB17                    ; keep looping after call
003344' 474 15 0 00 000000               SETO   ERFLG,                  ; error occurred

                                        ; Handle error printouts and scope looping

003345' 027 00 0 00 003372'             SCOPER  MA17                    ; print error message
003346' 254 00 0 00 003334'             JRST    TL17                    ; loop on error
003347' 254 00 0 00 003351'             JRST    TX17                    ; altmode exit
003350' 322 15 0 00 003340'             JUMPE   ERFLG,TA17              ; do next sstep table entry

                                        ; End of test

003351' 263 17 0 00 000000      TX17:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

                                        ; CC is set initially (KL last wrote CSR)

003352' 040000  000000          TS17:   TTABLE  (TSTART,0)              ; start up port
003353' 240002  000000                  TTABLE  (TCHECK,2)              ; check if completed
003354' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; CC is not set after port reads CSR

003355' 040000  000003                  TTABLE  (TSTART,3)              ; start up port
003356' 240002  000000                  TTABLE  (TCHECK,2)              ; check if completed
003357' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; CC is set after an EBUS PE

003360' 040000  000011                  TTABLE  (TSTART,11)             ; start up port
003361' 140000  003367'                 TTABLE  (TCALL,TS17PE)          ; cause an EBUS parity error
003362' 240002  000000                  TTABLE  (TCHECK,2)              ; check if completed
003363' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

```

```

                                       ; CC stays set if port writes CSR

 003364' 040000  000017                          TTABLE  (TSTART,17)           ; start up port
 003365' 240002  000000                          TTABLE  (TCHECK,2)            ; check if completed

 003366' 000000  000000                          TTABLE  (TLAST)

                                       ; Routine to cause an EBUS PE after the port has been started

 003367' 201 01 0 00 140010            TS17PE: MOVEI   1,GENEPE!MPRUN!SELLAR   ; get data to write to CSR
 003370' 260 17 0 00 002026*                   GO      LDCSR                   ; write it
 003371' 263 17 0 00 000000                    RTN                             ; return

                                       ; Error messages

 003372' 140000  012056'               MA17:   MSG!TXNOT![ASCIZ /Condition Code 'CC CSR Chng' not /]
 003373' 240000  003375'                       CALL!TXNOT!MA17LP               ; print subtest description
 003374' 000000000000#                         LAST!CALL!TXALL!TTPNT           ; print test data

 003375' 200 01 0 00 003336*           MA17LP: MOVE    1,TSTSUB                ; get subtest number
 003376' 256 00 0 01 003377'                   XCT     MA17L-1(1)              ; print message
 003377' 263 17 0 00 000000                    RTN                             ; return

 003400' 037 00 0 00 012065'           MA17L:  TMSG    <set initially (KL last wrote CSR)>
 003401' 037 00 0 00 012074'                   TMSG    <cleared after port reads CSR>
 003402' 037 00 0 00 012102'                   TMSG    <set after an EBUS PE>
 003403' 037 00 0 00 012107'                   TMSG    <still set after port writes CSR>

                                       ; Microcode:

                                       ;       CC is set initially (KL last wrote CSR)

 003404' 000000  010000                T17M:   MWORD   <ADDR=0,JMAP,J=1>                       ; 0
 003405' 000000  000040
 003406' 000151  000000                        MWORD   <CJP,J=5100,CENA,CCCC>                  ; 1
 003407' 000400  030060
 003410' 000252  000000                        MWORD   <JMAP,J=5200>                           ; 2
 003411' 000000  000040

                                       ;       CC is not set after port reads CSR

 003412' 000300  040000                        MWORD   <ADDR=3,JMAP,J=4>                       ; 3
 003413' 000000  000040
 003414' 000400  060040                        MWORD   <CJP,J=6,CENA,CCGC,D=1,SELE,MGC=40>     ; 4
 003415' 001400  015060
 003416' 000500  040000                        MWORD   <JMAP,J=4,D=1>                          ; 5
 003417' 001000  000040
 003420' 000600  000100                        MWORD   <CONT,SOA,OR,D=1,SELE,MGC=100>          ; 6
 003421' 431000  005340
 003422' 000752  000000                        MWORD   <CJP,J=5200,CENA,CCCC>                  ; 7
 003423' 000400  030060
 003424' 001051  000000                        MWORD   <JMAP,J=5100>                           ; 10
 003425' 000000  000040

```

```
                                        ;       CC is set after an EBUS PE

003426' 001100  120000                          MWORD   <ADDR=11,JMAP,J=12>                     ; 11
003427' 000000  000040
003430' 001200  140040                          MWORD   <CJP,J=14,CENA,CCGC,D=1,SELE,MGC=40>    ; 12
003431' 001400  015060
003432' 001300  120000                          MWORD   <JMAP,J=12,D=1>                         ; 13
003433' 001000  000040
003434' 001400  000100                          MWORD   <CONT,SOA,OR,D=1,SELE,MGC=100>          ; 14
003435' 431000  005340
003436' 001551  000000                          MWORD   <CJP,J=5100,CENA,CCCC>                  ; 15
003437' 000400  030060
003440' 001600  150000                          MWORD   <JMAP,J=15>                             ; 16
003441' 000000  000040

                                        ;       CC stays set if port writes CSR

003442' 001700  200000                          MWORD   <ADDR=17,JMAP,J=20>                     ; 17
003443' 000000  000040
003444' 002000  002004                          MWORD   <CONT,SDO,AND,D=1,OENA,SELE,MGC=4>      ; 20
003445' 741000  005340
003446' 002100  230040                          MWORD   <CJP,J=23,CENA,CCGC,D=1,SELE,MGC=40>    ; 21
003447' 001400  015060
003450' 002200  210000                          MWORD   <JMAP,J=21,D=1>                         ; 22
003451' 001000  000040
003452' 002300  000200                          MWORD   <CONT,SOA,OR,D=1,SELE,MGC=200>          ; 23
003453' 431000  005340
003454' 002452  000000                          MWORD   <CJP,J=5200,CENA,CCGC>                  ; 24
003455' 000400  010060
003456' 002551  000000                          MWORD   <JMAP,J=5100>                           ; 25
003457' 000000  000040

                                        ; Subroutine - Passed - Write 1 to bit 12 of CSR register

003460' 510000  000000                          MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>        ; 5100
003461' 742001  000340
003462' 510100  000000                          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5101
003463' 302001  000740
003464' 510200  260000                          MWORD   <LDCT,J=26>                             ; 5102
003465' 000000  000300
003466' 510351  030000                          MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>      ; 5103
003467' 102021  000220
003470' 510400  002004                          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5104
003471' 431020  005340
003472' 510551  070040                          MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>  ; 5105
003473' 001400  015060
003474' 510651  050000                          MWORD   <JMAP,J=5105,D=1>                       ; 5106
003475' 001000  000040
003476' 510700  000200                          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5107
003477' 431010  005340
003500' 511051  100000                          MWORD   <JMAP,J=5110,D=1>                       ; 5110
003501' 001000  000040

                                        ; Subroutine - Failed - Write 1 to bit 13 of CSR register

```

```
  2925  003502' 520000  000000              MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>        ; 5200
  2926  003503' 742001  000340
  2927  003504' 520100  000000              MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5201
  2928  003505' 302001  000740
  2929  003506' 520200  250000              MWORD   <LDCT,J=25>                             ; 5202
  2930  003507' 000000  000300
  2931  003510' 520352  030000              MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>      ; 5203
  2932  003511' 102021  000220
  2933  003512' 520400  002004              MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5204
  2934  003513' 431020  005340
  2935  003514' 520552  070040              MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>  ; 5205
  2936  003515' 001400  015060
  2937  003516' 520652  050000              MWORD   <JMAP,J=5205,D=1>                       ; 5206
  2938  003517' 001000  000040
  2939  003520' 520700  000200              MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5207
  2940  003521' 431010  005340
  2941  003522' 521052  100000              MWORD   <JMAP,J=5210,D=1>                       ; 5210
  2942  003523' 001000  000040
  2943  003524' 777777  777777              -1
```

```
                                   ;#*********************************************************************
                                   ;* TEST 20 - CCEbParErr Test
                                   ;
                                   ; Description:  Verify that condition  code  'CCEbParErr' can  be
                                   ;               tested  by  the  microsequencer  for  conditional
                                   ;               branching, and that the condition code can be set
                                   ;               and reset by the hardware conditions which should
                                   ;               do so.
                                   ;
                                   ; Procedure:    Port Clear
                                   ;               Start up port
                                   ;               UC> perform an operation and then verify that
                                   ;                      CCEbParErr is not asserted or asserted
                                   ;               UC> if passed - set bit 12 (DISABLE Complete)
                                   ;               UC> if failed - set bit 13 (ENABLE Complete)
                                   ;               Read CSR and determine test disposition
                                   ;
                                   ;               Repeat with several test cases:
                                   ;                 CC is cleared initially
                                   ;                 CC is set after an EBUS PE
                                   ;
                                   ; Failure:      ---
                                   ;#*********************************************************************

                                   ; Test data

003525' 254 00 0 00 003537'        TSTU20: JRST    TG20                     ; go start test
003526' 200403  000020                     MPROC!NDMP!ZMPROC!20             ; test mask
003527' 003604' 012116'                    T20M,,[ASCIZ ^CCEbParErr Test^]
003530' 012122' 012124'                    [EXP M6,MLAST!E16],,[EXP M7,E9,E1,E4,MLAST!E6]
003531' 000000  003677'                    TSTU21                           ; failure test table
003532' 000000  004070'                    TSTU22                           ;   ...
003533' 000000  004261'                    TSTU23
003534' 000000  004452'                    TSTU24
003535' 000000  004645'                    TSTU25
003536' 777777  777777                     -1

                                   ; Start test

003537' 201 00 0 00 000000'        TG20:   MOVEI   Z8                       ; get address of module start
003540' 260 17 0 00 003330*                GO      TRACE                    ; handle trace output
003541' 201 01 0 00 003604'                MOVEI   1,T20M                   ; set up microcode address
003542' 260 17 0 00 003332*                GO      TLOAD                    ; load/verify it
003543' 263 17 0 00 000000                 RTN                              ; failed - exit test

                                   ; Initialization

003544' 400 15 0 00 000000         TL20:   SETZ    ERFLG,                   ; clear error flag
003545' 201 06 0 00 003562'                MOVEI   6,TS20                   ; get test table address
003546' 402 00 0 00 003375*                SETZM   TSTSUB                   ; initialize subtest number
003547' 476 00 0 00 003337*                SETOM   TADDR                    ; clear start address
```

```

                                        ; Loop on test execute table entries

003550' 260 17 0 00 003340*             TA20:   GO      IPACLR                  ; clear port
003551' 260 17 0 00 003341*             TB20:   GO      TEXEC                   ; execute table entry
003552' 254 00 0 00 003561'                     JRST    TX20                    ; end of sstep table
003553' 254 00 0 00 003551'                     JRST    TB20                    ; keep looping after call
003554' 474 15 0 00 000000                      SETO    ERFLG,                  ; error occurred

                                        ; Handle error printouts and scope looping

003555' 027 00 0 00 003574'                     SCOPER  MA20                    ; print error message
003556' 254 00 0 00 003544'                     JRST    TL20                    ; loop on error
003557' 254 00 0 00 003561'                     JRST    TX20                    ; altmode exit
003560' 322 15 0 00 003550'                     JUMPE   ERFLG,TA20              ; do next sstep table entry

                                        ; End of test

003561' 263 17 0 00 000000              TX20:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

                                        ; CC is cleared initially

003562' 040000  000000                  TS20:   TTABLE  (TSTART,0)              ; start up port
003563' 240002  000000                          TTABLE  (TCHECK,2)              ; check if completed
003564' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; CC is set after an EBUS PE

003565' 040000  000003                          TTABLE  (TSTART,3)              ; start up port
003566' 140000  003571'                         TTABLE  (TCALL,TS20PE)          ; cause an EBUS parity error
003567' 240002  000000                          TTABLE  (TCHECK,2)              ; check if completed
003570' 000000  000000                          TTABLE  (TLAST)

                                        ; Routine to cause an EBUS PE after the port has been started

003571' 201 01 0 00 140010              TS20PE: MOVEI   1,GENEPE!MPRUN!SELLAR   ; get data to write to CSR
003572' 260 17 0 00 003370*                     GO      LDCSR                   ; write it
003573' 263 17 0 00 000000                      RTN                             ; return

                                        ; Error messages

003574' 140000  012131'                 MA20:   MSG!TXNOT![ASCIZ /Condition Code 'CC Eb Par Err' not /]
003575' 240000  003577'                         CALL!TXNOT!MA20LP               ; print subtest description
003576' 000000000000#                           LAST!CALL!TXALL!TTPNT           ; print test data

003577' 200 01 0 00 003546*             MA20LP: MOVE    1,TSTSUB                ; get subtest number
003600' 256 00 0 01 003601'                     XCT     MA20L-1(1)              ; print message
003601' 263 17 0 00 000000                      RTN                             ; return

003602' 037 00 0 00 011656'             MA20L:  TMSG    <cleared initially>
003603' 037 00 0 00 012102'                     TMSG    <set after an EBUS PE>
```

```

                                ; Microcode:

                                ;       CC is cleared initially

003604' 000000  010000          T20M:   MWORD   <ADDR=0,JMAP,J=1>                       ; 0
003605' 000000  000040
003606' 000152  000000                  MWORD   <CJP,J=5200,CENA,CCEB>                  ; 1
003607' 000400  040060
003610' 000251  000000                  MWORD   <JMAP,J=5100>                           ; 2
003611' 000000  000040

                                ;       CC is set after an EBUS PE

003612' 000300  040000                  MWORD   <ADDR=3,JMAP,J=4>                       ; 3
003613' 000000  000040
003614' 000400  060040                  MWORD   <CJP,J=6,CENA,CCGC,D=1,SELE,MGC=40>     ; 4
003615' 001400  015060
003616' 000500  040000                  MWORD   <JMAP,J=4,D=1>                          ; 5
003617' 001000  000040
003620' 000600  000100                  MWORD   <CONT,SOA,OR,D=1,SELE,MGC=100>          ; 6
003621' 431000  005340
003622' 000700  110000                  MWORD   <CJP,J=11,CENA,CCCC>                    ; 7
003623' 000400  030060
003624' 001000  070000                  MWORD   <JMAP,J=7>                              ; 10
003625' 000000  000040
003626' 001151  000000                  MWORD   <CJP,J=5100,CENA,CCEB>                  ; 11
003627' 000400  040060
003630' 001252  000000                  MWORD   <JMAP,J=5200>                           ; 12
003631' 000000  000040

                                ; Subroutine - Passed - Write 1 to bit 12 of CSR register

003632' 510000  000000                  MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>        ; 5100
003633' 742001  000340
003634' 510100  000000                  MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5101
003635' 302001  000740
003636' 510200  260000                  MWORD   <LDCT,J=26>                             ; 5102
003637' 000000  000300
003640' 510351  030000                  MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>      ; 5103
003641' 102021  000220
003642' 510400  002004                  MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5104
003643' 431020  005340
003644' 510551  070040                  MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>  ; 5105
003645' 001400  015060
003646' 510651  050000                  MWORD   <JMAP,J=5105,D=1>                       ; 5106
003647' 001000  000040
003650' 510700  000200                  MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5107
003651' 431010  005340
003652' 511051  100000                  MWORD   <JMAP,J=5110,D=1>                       ; 5110
003653' 001000  000040

                                ; Subroutine - Failed - Write 1 to bit 13 of CSR register

003654' 520000  000000                  MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>        ; 5200
```

```
  3104  003655' 742001  000340
  3105  003656' 520100  000000                        MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5201
  3106  003657' 302001  000740
  3107  003660' 520200  250000                        MWORD   <LDCT,J=25>                             ; 5202
  3108  003661' 000000  000300
  3109  003662' 520352  030000                        MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>      ; 5203
  3110  003663' 102021  000220
  3111  003664' 520400  002004                        MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5204
  3112  003665' 431020  005340
  3113  003666' 520552  070040                        MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>  ; 5205
  3114  003667' 001400  015060
  3115  003670' 520652  050000                        MWORD   <JMAP,J=5205,D=1>                       ; 5206
  3116  003671' 001000  000040
  3117  003672' 520700  000200                        MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5207
  3118  003673' 431010  005340
  3119  003674' 521052  100000                        MWORD   <JMAP,J=5210,D=1>                       ; 5210
  3120  003675' 001000  000040
  3121  003676' 777777  777777                        -1
```

```
                                          ;#*******************************************************************
                                          ;* TEST 21 - CCRcvrBufAFul Test
                                          ;
                                          ; Description:  Verify that condition code 'CCRcvrBufAFul' can be
                                          ;               tested  by  the  microsequencer  for  conditional
                                          ;               branching.
                                          ;
                                          ; Procedure:    Port Clear
                                          ;               Start up port
                                          ;               UC> perform an operation and then verify that
                                          ;                   CCRcvrBufAFul is not asserted
                                          ;               UC> if passed - set bit 12 (DISABLE Complete)
                                          ;               UC> if failed - set bit 13 (ENABLE Complete)
                                          ;               Read CSR and determine test disposition
                                          ;
                                          ;               Repeat with several test cases:
                                          ;                 CC is cleared initially (after resetting the
                                          ;                   bit in case the PB module is connected)
                                          ;                 CC is cleared after a CLRCCCODE command
                                          ;
                                          ; Failure:      ---
                                          ;#*******************************************************************

                                          ; Test data

 003677' 254 00 0 00 003711'              TSTU21: JRST    TG21                    ; go start test
 003700' 200403  000021                           MPROC!NDMP!ZMPROC!21            ; test mask
 003701' 003753' 012141'                          T21M,,[ASCIZ ^CCRcvrBufAFul Test^]
 003702' 012145' 012147'                          [EXP M6,MLAST!C13],,[EXP M7,MLAST!C7]
 003703' 000000  004070'                          TSTU22                          ; failure test table
 003704' 000000  004261'                          TSTU23                          ;   ...
 003705' 000000  004452'                          TSTU24
 003706' 000000  004645'                          TSTU25
 003707' 000000  005035'                          TSTU26
 003710' 777777  777777                           -1

                                          ; Start test

 003711' 201 00 0 00 000000'              TG21:   MOVEI   Z8                      ; get address of module start
 003712' 260 17 0 00 003540*                      GO      TRACE                   ; handle trace output
 003713' 201 01 0 00 003753'                      MOVEI   1,T21M                  ; set up microcode address
 003714' 260 17 0 00 003542*                      GO      TLOAD                   ; load/verify it
 003715' 263 17 0 00 000000                       RTN                             ; failed - exit test

                                          ; Initialization

 003716' 400 15 0 00 000000               TL21:   SETZ    ERFLG,                  ; clear error flag
 003717' 201 06 0 00 003734'                      MOVEI   6,TS21                  ; get test table address
 003720' 402 00 0 00 003577*                      SETZM   TSTSUB                  ; initialize subtest number
 003721' 476 00 0 00 003547*                      SETOM   TADDR                   ; clear start address
```

```

                                        ; Loop on test execute table entries

003722' 260 17 0 00 003550*             TA21:   GO      IPACLR                          ; clear port
003723' 260 17 0 00 003551*             TB21:   GO      TEXEC                           ; execute table entry
003724' 254 00 0 00 003733'                     JRST    TX21                            ; end of sstep table
003725' 254 00 0 00 003723'                     JRST    TB21                            ; keep looping after call
003726' 474 15 0 00 000000                      SETO    ERFLG,                          ; error occurred

                                        ; Handle error printouts and scope looping

003727' 027 00 0 00 003743'                     SCOPER  MA21                            ; print error message
003730' 254 00 0 00 003716'                     JRST    TL21                            ; loop on error
003731' 254 00 0 00 003733'                     JRST    TX21                            ; altmode exit
003732' 322 15 0 00 003722'                     JUMPE   ERFLG,TA21                      ; do next sstep table entry

                                        ; End of test

003733' 263 17 0 00 000000              TX21:   RTN                                     ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

                                        ; CC is cleared initially

003734' 040000  000030                  TS21:   TTABLE  (TSTART,30)                     ; start up port
003735' 240002  000000                          TTABLE  (TCHECK,2)                      ; check if completed
003736' 340000  000000                          TTABLE  (TEXIT)                         ; exit if error yet

                                        ; CC is cleared after a CLRCCCODE command

003737' 040000  000010                          TTABLE  (TSTART,10)                     ; start up port
003740' 240002  000000                          TTABLE  (TCHECK,2)                      ; check if completed
003741' 340000  000000                          TTABLE  (TEXIT)                         ; exit if error yet
003742' 000000  000000                          TTABLE  (TLAST)

                                        ; Error messages

003743' 140000  012151'                 MA21:   MSG!TXNOT![ASCIZ /Condition Code 'CC Rcvr BufA Ful' not /]
003744' 240000  003746'                         CALL!TXNOT!MA21LP               ; print subtest description
003745' 000000000000#                           LAST!CALL!TXALL!TTPNT           ; print test data

003746' 200 01 0 00 003720*             MA21LP: MOVE    1,TSTSUB                        ; get subtest number
003747' 256 00 0 01 003750'                     XCT     MA21L-1(1)                      ; print message
003750' 263 17 0 00 000000                      RTN                                     ; return

003751' 037 00 0 00 011656'             MA21L:  TMSG    <cleared initially>
003752' 037 00 0 00 012161'                     TMSG    <cleared after a CLRCCCODE command>
```

```

                                 ; Microcode:

                                 ;        CC is cleared initially

003753' 003000  000000           T21M:    MWORD   <ADDR=30,JZ>                            ; 30
003754' 000000  000000                    MWORD   <ADDR=0,CJS,J=40>                       ; 0
003755' 000000  400000                    MWORD   <CONT,SDO,OR,B=2,D=2,MGC=14>            ; 1
003756' 000000  000020                    MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELM,MGC=40>  ; 2
003757' 000100  000014                    MWORD   <CONT,SDO,D=1,SELP,MGC=274>             ; 3
003760' 732001  000340                    MWORD   <CONT,D=1>                              ; 4
003761' 000200  002040                    MWORD   <CJP,J=5200,CENA,CCAF>                  ; 5
003762' 431020  002340                    MWORD   <JMAP,J=5100>                           ; 6
003763' 000300  000274
003764' 701000  001340
003765' 000400  000000
003766' 001000  000340
003767' 000552  000000
003770' 000400  050060
003771' 000651  000000
003772' 000000  000040

                                 ;        CC is cleared after a CLRCCCODE command
003773' 001000  110000                    MWORD   <ADDR=10,JMAP,J=11>                     ; 10
003774' 000000  000040                    MWORD   <CJS,J=40>                              ; 11
003775' 001100  400000                    MWORD   <CONT,SELM,MGC=20>                      ; 12
003776' 000000  000020                    MWORD   <CONT,D=1>                              ; 13
003777' 001200  000020                    MWORD   <CJP,J=5200,CENA,CCAF>                  ; 14
004000' 000000  002340                    MWORD   <JMAP,J=5100>                           ; 15
004001' 001300  000000
004002' 001000  000340
004003' 001452  000000
004004' 000400  050060
004005' 001551  000000
004006' 000000  000040

                                 ; Subroutine - Determine if a PILA module is connected - if not exit test
004007' 004000  420000                    MWORD   <ADDR=40,CJP,J=42,CENA,CCAF>            ; 40
004010' 000400  050060                    MWORD   <CRTN>                                  ; 41
004011' 004100  000000                    MWORD   <CJP,J=44,CENA,CCAF>                    ; 42
004012' 000000  000240                    MWORD   <CRTN>                                  ; 43
004013' 004200  440000                    MWORD   <CJP,J=5100,CENA,CCAF>                  ; 44
004014' 000400  050060                    MWORD   <CRTN>                                  ; 45
004015' 004300  000000
004016' 000000  000240
004017' 004451  000000
004020' 000400  050060
004021' 004500  000000
004022' 000000  000240

                                 ; Subroutine - Passed - Write 1 to bit 12 of CSR register

004023' 510000  000000                    MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>        ; 5100
```

```
3275  004024' 742001  000340           MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                 ; 5101
3276  004025' 510100  000000
3277  004026' 302001  000740           MWORD   <LDCT,J=26>                                 ; 5102
3278  004027' 510200  260000
3279  004030' 000000  000300           MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>          ; 5103
3280  004031' 510351  030000
3281  004032' 102021  000220           MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>       ; 5104
3282  004033' 510400  002004
3283  004034' 431020  005340           MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>      ; 5105
3284  004035' 510551  070040
3285  004036' 001400  015060           MWORD   <JMAP,J=5105,D=1>                           ; 5106
3286  004037' 510651  050000
3287  004040' 001000  000040           MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>          ; 5107
3288  004041' 510700  000200
3289  004042' 431010  005340           MWORD   <JMAP,J=5110,D=1>                           ; 5110
3290  004043' 511051  100000
3291  004044' 001000  000040
3292
3293                          ; Subroutine - Failed - Write 1 to bit 13 of CSR register
3294
3295  004045' 520000  000000           MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>            ; 5200
3296  004046' 742001  000340
3297  004047' 520100  000000           MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                 ; 5201
3298  004050' 302001  000740
3299  004051' 520200  250000           MWORD   <LDCT,J=25>                                 ; 5202
3300  004052' 000000  000300
3301  004053' 520352  030000           MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>          ; 5203
3302  004054' 102021  000220
3303  004055' 520400  002004           MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>       ; 5204
3304  004056' 431020  005340
3305  004057' 520552  070040           MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>      ; 5205
3306  004060' 001400  015060
3307  004061' 520652  050000           MWORD   <JMAP,J=5205,D=1>                           ; 5206
3308  004062' 001000  000040
3309  004063' 520700  000200           MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>          ; 5207
3310  004064' 431010  005340
3311  004065' 521052  100000           MWORD   <JMAP,J=5210,D=1>                           ; 5210
3312  004066' 001000  000040
3313  004067' 777777  777777           -1
```

```
                                           ;#*********************************************************************
                                           ;* TEST 22 - CCRcvrBufBFul Test
                                           ;
                                           ; Description:  Verify that condition code 'CCRcvrBufBFul' can be
                                           ;               tested  by  the  microsequencer  for  conditional
                                           ;               branching.
                                           ;
                                           ; Procedure:    Port Clear
                                           ;               Start up port
                                           ;               UC> perform an operation and then verify that
                                           ;                   CCRcvrBufBFul is not asserted
                                           ;               UC> if passed - set bit 12 (DISABLE Complete)
                                           ;               UC> if failed - set bit 13 (ENABLE Complete)
                                           ;               Read CSR and determine test disposition
                                           ;
                                           ;               Repeat with several test cases:
                                           ;                 CC is cleared initially (after resetting the
                                           ;                   bit in case the PB module is connected)
                                           ;                 CC is cleared after a CLRCCCODE command
                                           ;
                                           ; Failure:      ---
                                           ;#*********************************************************************

                                           ; Test data

004070' 254 00 0 00 004102'                TSTU22: JRST    TG22                    ; go start test
004071' 200403  000022                             MPROC!NDMP!ZMPROC!22            ; test mask
004072' 004144' 012170'                            T22M,,[ASCIZ ^CCRcvrBufBFul Test^]
004073' 012145' 012147'                            [EXP M6,MLAST!C13],,[EXP M7,MLAST!C7]
004074' 000000  004261'                            TSTU23                          ; failure test table
004075' 000000  004452'                            TSTU24                          ;   ...
004076' 000000  004645'                            TSTU25
004077' 000000  005035'                            TSTU26
004100' 000000  005216'                            TSTU27
004101' 777777  777777                             -1

                                           ; Start test

004102' 201 00 0 00 000000'                TG22:   MOVEI   Z8                      ; get address of module start
004103' 260 17 0 00 003712*                        GO      TRACE                   ; handle trace output
004104' 201 01 0 00 004144'                        MOVEI   1,T22M                  ; set up microcode address
004105' 260 17 0 00 003714*                        GO      TLOAD                   ; load/verify it
004106' 263 17 0 00 000000                         RTN                             ; failed - exit test

                                           ; Initialization

004107' 400 15 0 00 000000                 TL22:   SETZ    ERFLG,                  ; clear error flag
004110' 201 06 0 00 004125'                        MOVEI   6,TS22                  ; get test table address
004111' 402 00 0 00 003746*                        SETZM   TSTSUB                  ; initialize subtest number
004112' 476 00 0 00 003721*                        SETOM   TADDR                   ; clear start address
```

```
                                          ; Loop on test execute table entries

004113' 260 17 0 00 003722*               TA22:   GO      IPACLR                  ; clear port
004114' 260 17 0 00 003723*               TB22:   GO      TEXEC                   ; execute table entry
004115' 254 00 0 00 004124'                        JRST    TX22                    ; end of sstep table
004116' 254 00 0 00 004114'                        JRST    TB22                    ; keep looping after call
004117' 474 15 0 00 000000                         SETO    ERFLG,                  ; error occurred

                                          ; Handle error printouts and scope looping

004120' 027 00 0 00 004134'                       SCOPER  MA22                    ; print error message
004121' 254 00 0 00 004107'                       JRST    TL22                    ; loop on error
004122' 254 00 0 00 004124'                       JRST    TX22                    ; altmode exit
004123' 322 15 0 00 004113'                       JUMPE   ERFLG,TA22              ; do next sstep table entry

                                          ; End of test

004124' 263 17 0 00 000000                TX22:   RTN                             ; return

                                          ; Test Execute Table, as:  (CMD,parameters)

                                          ; CC is cleared initially

004125' 040000  000030                    TS22:   TTABLE  (TSTART,30)             ; start up port
004126' 240002  000000                            TTABLE  (TCHECK,2)              ; check if completed
004127' 340000  000000                            TTABLE  (TEXIT)                 ; exit if error yet

                                          ; CC is cleared after a CLRCCCODE command

004130' 040000  000010                            TTABLE  (TSTART,10)             ; start up port
004131' 240002  000000                            TTABLE  (TCHECK,2)              ; check if completed
004132' 340000  000000                            TTABLE  (TEXIT)                 ; exit if error yet
004133' 000000  000000                            TTABLE  (TLAST)

                                          ; Error messages

004134' 140000  012174'                   MA22:   MSG!TXNOT![ASCIZ /Condition Code 'CC Rcvr BufB Ful' not /]
004135' 240000  004137'                           CALL!TXNOT!MA22LP               ; print subtest description
004136' 000000000000#                             LAST!CALL!TXALL!TTPNT           ; print test data

004137' 200 01 0 00 004111*               MA22LP: MOVE    1,TSTSUB                ; get subtest number
004140' 256 00 0 01 004141'                       XCT     MA22L-1(1)              ; print message
004141' 263 17 0 00 000000                        RTN                             ; return

004142' 037 00 0 00 011656'               MA22L:  TMSG    <cleared initially>
004143' 037 00 0 00 012161'                       TMSG    <cleared after a CLRCCCODE command>
```

```

                                        ; Microcode:

                                        ;       CC is cleared initially

004144' 003000  000000                  T22M:   MWORD   <ADDR=30,JZ>                            ; 30
004145' 000000  000000
004146' 000000  400000                          MWORD   <ADDR=0,CJS,J=40>                       ; 0
004147' 000000  000020
004150' 000100  000014                          MWORD   <CONT,SDO,OR,B=2,D=2,MGC=14>            ; 1
004151' 732001  000340
004152' 000200  002040                          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELM,MGC=40>  ; 2
004153' 431020  002340
004154' 000300  000274                          MWORD   <CONT,SDO,D=1,SELP,MGC=274>             ; 3
004155' 701000  001340
004156' 000400  000000                          MWORD   <CONT,D=1>                              ; 4
004157' 001000  000340
004160' 000552  000000                          MWORD   <CJP,J=5200,CENA,CCBF>                  ; 5
004161' 000400  060060
004162' 000651  000000                          MWORD   <JMAP,J=5100>                           ; 6
004163' 000000  000040

                                        ;       CC is cleared after a CLRCCCODE command

004164' 001000  110000                          MWORD   <ADDR=10,JMAP,J=11>                     ; 10
004165' 000000  000040
004166' 001100  400000                          MWORD   <CJS,J=40>                              ; 11
004167' 000000  000020
004170' 001200  000020                          MWORD   <CONT.SELM,MGC=20>                      ; 12
004171' 000000  002340
004172' 001300  000000                          MWORD   <CONT,D=1>                              ; 13
004173' 001000  000340
004174' 001452  000000                          MWORD   <CJP,J=5200,CENA,CCBF>                  ; 14
004175' 000400  060060
004176' 001551  000000                          MWORD   <JMAP,J=5100>                           ; 15
004177' 000000  000040

                                        ; Subroutine - Determine if a PILA module is connected - if not exit test

004200' 004000  420000                          MWORD   <ADDR=40,CJP,J=42,CENA,CCAF>            ; 40
004201' 000400  050060
004202' 004100  000000                          MWORD   <CRTN>                                  ; 41
004203' 000000  000240
004204' 004200  440000                          MWORD   <CJP,J=44,CENA,CCAF>                    ; 42
004205' 000400  050060
004206' 004300  000000                          MWORD   <CRTN>                                  ; 43
004207' 000000  000240
004210' 004451  000000                          MWORD   <CJP,J=5100,CENA,CCAF>                  ; 44
004211' 000400  050060
004212' 004500  000000                          MWORD   <CRTN>                                  ; 45
004213' 000000  000240

                                        ; Subroutine - Passed - Write 1 to bit 12 of CSR register

004214' 510000  000000                          MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>        ; 5100
```

```
004215' 742001  000340
004216' 510100  000000                  MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                ; 5101
004217' 302001  000740
004220' 510200  260000                  MWORD   <LDCT,J=26>                                ; 5102
004221' 000000  000300
004222' 510351  030000                  MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>         ; 5103
004223' 102021  000220
004224' 510400  002004                  MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>      ; 5104
004225' 431020  005340
004226' 510551  070040                  MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>     ; 5105
004227' 001400  015060
004230' 510651  050000                  MWORD   <JMAP,J=5105,D=1>                          ; 5106
004231' 001000  000040
004232' 510700  000200                  MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>         ; 5107
004233' 431010  005340
004234' 511051  100000                  MWORD   <JMAP,J=5110,D=1>                          ; 5110
004235' 001000  000040

                                ; Subroutine - Failed - Write 1 to bit 13 of CSR register

004236' 520000  000000                  MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>           ; 5200
004237' 742001  000340
004240' 520100  000000                  MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                ; 5201
004241' 302001  000740
004242' 520200  250000                  MWORD   <LDCT,J=25>                                ; 5202
004243' 000000  000300
004244' 520352  030000                  MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>         ; 5203
004245' 102021  000220
004246' 520400  002004                  MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>      ; 5204
004247' 431020  005340
004250' 520552  070040                  MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>     ; 5205
004251' 001400  015060
004252' 520652  050000                  MWORD   <JMAP,J=5205,D=1>                          ; 5206
004253' 001000  000040
004254' 520700  000200                  MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>         ; 5207
004255' 431010  005340
004256' 521052  100000                  MWORD   <JMAP,J=5210,D=1>                          ; 5210
004257' 001000  000040
004260' 777777  777777                  -1
```

```

                                  ;#*****************************************************************
                                  ;* TEST 23 - CCXmtrAttn Test
                                  ;
                                  ; Description:  Verify that condition code  'CCXmtrAttn'  can  be
                                  ;               tested  by  the  microsequencer  for  conditional
                                  ;               branching.
                                  ;
                                  ; Procedure:    Port Clear
                                  ;               Start up port
                                  ;               UC> perform an operation and then verify that
                                  ;                   CCXmtrAttn is not asserted
                                  ;               UC> if passed - set bit 12 (DISABLE Complete)
                                  ;               UC> if failed - set bit 13 (ENABLE Complete)
                                  ;               Read CSR and determine test disposition
                                  ;
                                  ;               Repeat with several test cases:
                                  ;                 CC is cleared initially (after resetting the
                                  ;                   bit in case the PB module is connected)
                                  ;                 CC is cleared after a CLRCCCODE command
                                  ;
                                  ; Failure:      ---
                                  ;#*****************************************************************

                                  ; Test data

004261' 254 00 0 00 004273'       TSTU23: JRST    TG23                    ; go start test
004262' 200403  000023                    MPROC!NDMP!ZMPROC!23            ; test mask
004263' 004335' 012204'                   T23M,,[ASCIZ ^CCXmtrAttn Test^]
004264' 012145' 012014'                   [EXP M6,MLAST!C13],,[EXP MLAST!M7]
004265' 000000  004452'                   TSTU24                          ; failure test table
004266' 000000  004645'                   TSTU25                          ;   ...
004267' 000000  005035'                   TSTU26
004270' 000000  005216'                   TSTU27
004271' 000000  005512'                   TSTU30
004272' 777777  777777                    -1

                                  ; Start test

004273' 201 00 0 00 000000'       TG23:   MOVEI   Z8                      ; get address of module start
004274' 260 17 0 00 004103*               GO      TRACE                   ; handle trace output
004275' 201 01 0 00 004335'               MOVEI   1,T23M                  ; set up microcode address
004276' 260 17 0 00 004105*               GO      TLOAD                   ; load/verify it
004277' 263 17 0 00 000000                RTN                             ; failed - exit test

                                  ; Initialization

004300' 400 15 0 00 000000        TL23:   SETZ    ERFLG,                  ; clear error flag
004301' 201 06 0 00 004316'               MOVEI   6,TS23                  ; get test table address
004302' 402 00 0 00 004137*               SETZM   TSTSUB                  ; initialize subtest number
004303' 476 00 0 00 004112*               SETOM   TADDR                   ; clear start address
```

```

                                        ; Loop on test execute table entries

004304' 260 17 0 00 004113*     TA23:   GO      IPACLR                  ; clear port
004305' 260 17 0 00 004114*     TB23:   GO      TEXEC                   ; execute table entry
004306' 254 00 0 00 004315'              JRST   TX23                    ; end of sstep table
004307' 254 00 0 00 004305'              JRST   TB23                    ; keep looping after call
004310' 474 15 0 00 000000               SETO   ERFLG,                  ; error occurred

                                        ; Handle error printouts and scope looping

004311' 027 00 0 00 004325'             SCOPER  MA23                    ; print error message
004312' 254 00 0 00 004302'             JRST    TL23                    ; loop on error
004313' 254 00 0 00 004315'             JRST    TX23                    ; altmode exit
004314' 322 15 0 00 004304'             JUMPE   ERFLG,TA23              ; do next sstep table entry

                                        ; End of test

004315' 263 17 0 00 000000      TX23:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

                                        ; CC is cleared initially

004316' 040000  000030          TS23:   TTABLE  (TSTART,30)             ; start up port
004317' 240002  000000                  TTABLE  (TCHECK,2)              ; check if completed
004320' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; CC is cleared after a CLRCCCODE command

004321' 040000  000010                  TTABLE  (TSTART,10)             ; start up port
004322' 240002  000000                  TTABLE  (TCHECK,2)              ; check if completed
004323' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet
004324' 000000  000000                  TTABLE  (TLAST)

                                        ; Error messages

004325' 140000  012210'         MA23:   MSG!TXNOT![ASCIZ /Condition Coc 'CC Xmtr Attn' not /]
004326' 240000  004330'                 CALL!TXNOT!MA23LP               ; print subtest description
004327' 000000000000#                   LAST!CALL!TXALL!TTPNT           ; print test data

004330' 200 01 0 00 004302*     MA23LP: MOVE    1,TSTSUB                ; get subtest number
004331' 256 00 0 01 004332'             XCT     MA23L-1(1)              ; print message
004332' 263 17 0 00 000000              RTN                             ; return

004333' 037 00 0 00 011656'     MA23L:  TMSG    <cleared initially>
004334' 037 00 0 00 012161'             TMSG    <cleared after a CLRCCCODE command>
```

```

                                  ; Microcode:

                                  ;       CC is cleared initially

004335' 003000  000000            T23M:   MWORD   <ADDR=30,JZ>                           ; 30
004336' 000000  000000
004337' 000000  400000                    MWORD   <ADDR=0,CJS,J=40>                      ; 0
004340' 000000  000020
004341' 000100  000050                    MWORD   <CONT,D=1,SELP,MGC=50>                 ; 1
004342' 001000  001340
004343' 000200  000000                    MWORD   <CONT,D=1>                             ; 2
004344' 001000  000340
004345' 000300  000000                    MWORD   <CONT,D=1>                             ; 3
004346' 001000  000340
004347' 000400  000000                    MWORD   <CONT,D=1>                             ; 4
004350' 001000  000340
004351' 000552  000000                    MWORD   <CJP,J=5200,CENA,CCXA>                 ; 5
004352' 000400  070060
004353' 000651  000000                    MWORD   <JMAP,J=5100>                          ; 6
004354' 000000  000040

                                  ;       CC is cleared after a CLRCCODE command

004355' 001000  110000                    MWORD   <ADDR=10,JMAP,J=11>                    ; 10
004356' 000000  000040
004357' 001100  400000                    MWORD   <CJS,J=40>                             ; 11
004360' 000000  000020
004361' 001200  000020                    MWORD   <CONT,SELM,MGC=20>                     ; 12
004362' 000000  002340
004363' 001300  000000                    MWORD   <CONT,D=1>                             ; 13
004364' 001000  000340
004365' 001452  000000                    MWORD   <CJP,J=5200,CENA,CCAF>                 ; 14
004366' 000400  050060
004367' 001551  000000                    MWORD   <JMAP,J=5100>                          ; 15
004370' 000000  000040

                                  ; Subroutine - Determine if a PILA module is connected - if not exit test

004371' 004000  420000                    MWORD   <ADDR=40,CJP,J=42,CENA,CCAF>           ; 40
004372' 000400  050060
004373' 004100  000000                    MWORD   <CRTN>                                 ; 41
004374' 000000  000240
004375' 004200  440000                    MWORD   <CJP,J=44,CENA,CCAF>                   ; 42
004376' 000400  050060
004377' 004300  000000                    MWORD   <CRTN>                                 ; 43
004400' 000000  000240
004401' 004451  000000                    MWORD   <CJP,J=5100,CENA,CCAF>                 ; 44
004402' 000400  050060
004403' 004500  000000                    MWORD   <CRTN>                                 ; 45
004404' 000000  000240

                                  ; Subroutine - Passed - Write 1 to bit 12 of CSR register

004405' 510000  000000                    MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>       ; 5100
```

```
004406' 742001  000340
004407' 510100  000000          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                     ; 5101
004410' 302001  000740
004411' 510200  260000          MWORD   <LDCT,J=26>                                     ; 5102
004412' 000000  000300
004413' 510351  030000          MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>              ; 5103
004414' 102021  000220
004415' 510400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>           ; 5104
004416' 431020  005340
004417' 510551  070040          MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>          ; 5105
004420' 001400  015060
004421' 510651  050000          MWORD   <JMAP,J=5105,D=1>                               ; 5106
004422' 001000  000040
004423' 510700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>              ; 5107
004424' 431010  005340
004425' 511051  100000          MWORD   <JMAP,J=5110,D=1>                               ; 5110
004426' 001000  000040

                        ; Subroutine - Failed - Write 1 to bit 13 of CSR register

004427' 520000  000000          MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>                ; 5200
004430' 742001  000340
004431' 520100  000000          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                     ; 5201
004432' 302001  000740
004433' 520200  250000          MWORD   <LDCT,J=25>                                     ; 5202
004434' 000000  000300
004435' 520352  030000          MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>              ; 5203
004436' 102021  000220
004437' 520400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>           ; 5204
004440' 431020  005340
004441' 520552  070040          MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>          ; 5205
004442' 001400  015060
004443' 520652  050000          MWORD   <JMAP,J=5205,D=1>                               ; 5206
004444' 001000  000040
004445' 520700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>              ; 5207
004446' 431010  005340
004447' 521052  100000          MWORD   <JMAP,J=5210,D=1>                               ; 5210
004450' 001000  000040
004451' 777777  777777          -1
```

```
                                    ;#*********************************************************************
                                    ;* TEST 24 - CCEbusRqst Test
                                    ;
                                    ; Description:  Verify that condition code 'CCEbusRqst'  can  be
                                    ;               tested  by  the  microsequencer  for  conditional
                                    ;               branching, and that the condition code can be set
                                    ;               and reset by the hardware conditions which should
                                    ;               do so.
                                    ;
                                    ; Procedure:    Port Clear
                                    ;               Start up port
                                    ;               UC> perform an operation and then verify that
                                    ;                   CCEbusRqst is not asserted or asserted
                                    ;               UC> if passed - set bit 12 (DISABLE Complete)
                                    ;               UC> if failed - set bit 13 (ENABLE Complete)
                                    ;               Read CSR and determine test disposition
                                    ;
                                    ;               Repeat with several test cases:
                                    ;                 CC is cleared initially
                                    ;                 CC sets when KL tries to to a DATAI
                                    ;                 CC sets when KL tries to to a DATAO
                                    ;                 CC clears after KL's DATAI times out
                                    ;                 CC clears after KL's DATAO times out
                                    ;
                                    ; Failure:      ---
                                    ;#*********************************************************************

                                    ; Test data

004452' 254 00 0 00 004464'         TSTU24: JRST    TG24                    ; go start test
004453' 200403  000024                      MPROC!NDMP!ZMPROC!24            ; test mask
004454' 004552' 012217'                     T24M,,[ASCIZ ^CCEbusRqst Test^]
004455' 012223' 012225'                     [EXP M6,MLAST!E2],,[EXP M7,E1,MLAST!E17]
004456' 000000  004645'                     TSTU25                          ; failure test table
004457' 000000  005035'                     TSTU26                          ;   ...
004460' 000000  005216'                     TSTU27
004461' 000000  005512'                     TSTU30
004462' 000000  005747'                     TSTU31
004463' 777777  777777                      -1

                                    ; Start test

004464' 201 00 0 00 000000'         TG24:   MOVEI   Z8                      ; get address of module start
004465' 260 17 0 00 004274*                 GO      TRACE                   ; handle trace output
004466' 201 01 0 00 004552'                 MOVEI   1,T24M                  ; set up microcode address
004467' 260 17 0 00 004276*                 GO      TLOAD                   ; load/verify it
004470' 263 17 0 00 000000                  RTN                             ; failed - exit test
```

```

                                  ; Initialization

004471' 400 15 0 00 000000        TL24:  SETZ    ERFLG,                  ; clear error flag
004472' 201 06 0 00 004507'              MOVEI   6,TS24                  ; get test table address
004473' 402 00 0 00 004330*              SETZM   TSTSUB                  ; initialize subtest number
004474' 476 00 0 00 004303*              SETOM   TADDR                   ; clear start address

                                  ; Loop on test execute table entries

004475' 260 17 0 00 004304*       TA24:  GO      IPACLR                  ; clear port
004476' 260 17 0 00 004305*       TB24:  GO      TEXEC                   ; execute table entry
004477' 254 00 0 00 004506'               JRST   TX24                    ; end of sstep table
004500' 254 00 0 00 004476'               JRST   TB24                    ; keep looping after call
004501' 474 15 0 00 000000                SETO   ERFLG,                  ; error occurred

                                  ; Handle error printouts and scope looping

004502' 027 00 0 00 004537'              SCOPER  MA24                    ; print error message
004503' 254 00 0 00 004471'              JRST    TL24                    ; loop on error
004504' 254 00 0 00 004506'              JRST    TX24                    ; altmode exit
004505' 322 15 0 00 004475'              JUMPE   ERFLG,TA24              ; do next sstep table entry

                                  ; End of test

004506' 263 17 0 00 000000        TX24:  RTN                             ; return

                                  ; Test Execute Table, as:  (CMD,parameters)

                                  ; CC is cleared initially

004507' 040000  000000            TS24:  TTABLE  (TSTART,0)              ; start up port
004510' 240002  000000                   TTABLE  (TCHECK,2)              ; check if completed
004511' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

                                  ; CC sets when KL tries to to a DATAI

004512' 040000  000003                   TTABLE  (TSTART,3)              ; start up port
004513' 140000  004533'                  TTABLE  (TCALL,TS24DI)          ; do a DATAI
004514' 240002  000000                   TTABLE  (TCHECK,2)              ; check if completed
004515' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

                                  ; CC sets when KL tries to to a DATAO

004516' 040000  000003                   TTABLE  (TSTART,3)              ; start up port
004517' 140000  004535'                  TTABLE  (TCALL,TS24DO)          ; do a DATAO
004520' 240002  000000                   TTABLE  (TCHECK,2)              ; check if completed
004521' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

```

```


                                 ; CC clears after KL's DATAI times out

004522' 040000  000006                    TTABLE  (TSTART,6)                  ; start up port
004523' 140000  004533'                   TTABLE  (TCALL,TS24DI)              ; do a DATAI
004524' 240002  000000                    TTABLE  (TCHECK,2)                  ; check if completed
004525' 340000  000000                    TTABLE  (TEXIT)                     ; exit if error yet

                                 ; CC clears after KL's DATAO times out

004526' 040000  000006                    TTABLE  (TSTART,6)                  ; start up port
004527' 140000  004535'                   TTABLE  (TCALL,TS24DO)              ; do a DATAO
004530' 240002  000000                    TTABLE  (TCHECK,2)                  ; check if completed
004531' 340000  000000                    TTABLE  (TEXIT)                     ; exit if error yet
004532' 000000  000000                    TTABLE  (TLAST)

                                 ; Routines to do a DATAI/O

004533' 260 17 0 00 002313*      TS24DI:  GO      RDLAR                       ; read LAR
004534' 263 17 0 00 000000                RTN                                 ; return

004535' 260 17 0 00 002316*      TS24DO:  GO      LDRAR                       ; load RAR
004536' 263 17 0 00 000000                RTN                                 ; return

                                 ; Error messages

004537' 140000  012230'          MA24:    MSG!TXNOT![ASCIZ /Condition Code 'CC EBUS Rqst' not /]
004540' 240000  004542'                   CALL!TXNOT!MA24LP                   ; print subtest description
004541' 000000000000#                     LAST!CALL!TXALL!TTPNT               ; print test data

004542' 200 01 0 00 004473*      MA24LP:  MOVE    1,TSTSUB                    ; get subtest number
004543' 256 00 0 01 004544'               XCT     MA24L-1(1)                  ; print message
004544' 263 17 0 00 000000                RTN                                 ; return

004545' 037 00 0 00 011656'      MA24L:   TMSG    <cleared initially>
004546' 037 00 0 00 012237'               TMSG    <set when KL tries to to a DATAI>
004547' 037 00 0 00 012246'               TMSG    <set when KL tries to to a DATAO>
004550' 037 00 0 00 012255'               TMSG    <cleared after KL's DATAI times out>
004551' 037 00 0 00 012264'               TMSG    <cleared after KL's DATAO times out>
```

```
                                        ; Microcode:

                                        ;       CC is cleared initially

004552' 000000  010000                  T24M:   MWORD   <ADDR=0,JMAP,J=1>                       ; 0
004553' 000000  000040
004554' 000152  000000                          MWORD   <CJP,J=5200,CENA,CCER>                  ; 1
004555' 000400  100060
004556' 000251  000000                          MWORD   <JMAP,J=5100>                           ; 2
004557' 000000  000040

                                        ;       CC sets when KL tries to to a DATAI/DATAO

004560' 000300  040000                          MWORD   <ADDR=3,JMAP,J=4>                       ; 3
004561' 000000  000040
004562' 000451  000000                          MWORD   <CJP,J=5100,CENA,CCER>                  ; 4
004563' 000400  100060
004564' 000500  040000                          MWORD   <JMAP,J=4>                              ; 5
004565' 000000  000040

                                        ;       CC clears after KL's DATAI/DATAO times out

004566' 000600  070000                          MWORD   <ADDR=6,JMAP,J=7>                       ; 6
004567' 000000  000040
004570' 000700  110000                          MWORD   <CJP,J=11,CENA,CCER>                    ; 7
004571' 000400  100060
004572' 001000  070000                          MWORD   <JMAP,J=7>                              ; 10
004573' 000000  000040
004574' 001100  110000                          MWORD   <CJP,J=11,CENA,CCER>                    ; 11
004575' 000400  100060
004576' 001251  000000                          MWORD   <JMAP,J=5100>                           ; 12
004577' 000000  000040

                                        ; Subroutine - Passed - Write 1 to bit 12 of CSR register

004600' 510000  000000                          MWORD   <ADDR=5100,CONT,SD0,AND,B=2,D=2>        ; 5100
004601' 742001  000340
004602' 510100  000000                          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5101
004603' 302001  000740
004604' 510200  260000                          MWORD   <LDCT,J=26>                             ; 5102
004605' 000000  000300
004606' 510351  030000                          MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>      ; 5103
004607' 102021  000220
004610' 510400  002004                          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5104
004611' 431020  005340
004612' 510551  070040                          MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>  ; 5105
004613' 001400  015060
004614' 510651  050000                          MWORD   <JMAP,J=5105,D=1>                       ; 5106
004615' 001000  000040
004616' 510700  000200                          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5107
004617' 431010  005340
004620' 511051  100000                          MWORD   <JMAP,J=5110,D=1>                       ; 5110
004621' 001000  000040

```

```
                                        ; Subroutine - Failed - Write 1 to bit 13 of CSR register

004622' 520000  000000                          MWORD   <ADDR=5200,CONT SDO,AND,B=2,D=2>        ; 5200
004623' 742001  000340
004624' 520100  000000                          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5201
004625' 302001  000740
004626' 520200  250000                          MWORD   <LDCT,J=25>                             ; 5202
004627' 000000  000300
004630' 520352  030000                          MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>      ; 5203
004631' 102021  000220
004632' 520400  002004                          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5204
004633' 431020  005340
004634' 520552  070040                          MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>  ; 5205
004635' 001400  015060
004636' 520652  050000                          MWORD   <JMAP,J=5205,D=1>                       ; 5206
004637' 001000  000040
004640' 520700  000200                          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5207
004641' 431010  005340
004642' 521052  100000                          MWORD   <JMAP,J=5210,D=1>                       ; 5210
004643' 001000  000040
004644' 777777  777777                          -1
```

```

                                    ;#***********************************************************************
                                    ;* TEST 25 - CCIntrActive Test
                                    ;
                                    ; Description:  Verify that condition code  'CCIntrActive' can be
                                    ;               tested  by  the  microsequencer  for  conditional
                                    ;               branching, and that the condition code can be set
                                    ;               and reset by the hardware conditions which should
                                    ;               do so.
                                    ;
                                    ; Procedure:    Port Clear
                                    ;               Start up port
                                    ;               UC> perform an operation and then verify that
                                    ;                   CCIntrActive is not asserted or asserted
                                    ;               UC> if passed - set bit 12 (DISABLE Complete)
                                    ;               UC> if failed - set bit 13 (ENABLE Complete)
                                    ;               Read CSR and determine test disposition
                                    ;
                                    ;               Repeat with several test cases:
                                    ;                 CC is cleared initially
                                    ;                 CC sets when an interrupt is pending
                                    ;                 CC clears after an interrupt has been handled
                                    ;
                                    ; Failure:      ---
                                    ;#***********************************************************************

                                    ; Test data

004645' 254 00 0 00 004657'         TSTU25: JRST    TG25                    ; go start test
004646' 200403  000025                      MPROC!NDMP!ZMPROC!25            ; test mask
004647' 004742' 012273'                     T25M,,[ASCIZ ^CCIntrActive Test^]
004650' 012277' 012301'                     [EXP M6,MLAST!E14],,[EXP M7,E15,E7,MLAST!E6]
004651' 000000  005035'                     TSTU26                          ; failure test table
004652' 000000  005216'                     TSTU27                          ;   ...
004653' 000000  005512'                     TSTU30
004654' 000000  005747'                     TSTU31
004655' 000000  006307'                     TSTU32
004656' 777777  777777                      -1

                                    ; Start test

004657' 201 00 0 00 000000'         TG25:   MOVEI   Z8                      ; get address of module start
004660' 260 17 0 00 004465*                 GO      TRACE                   ; handle trace output
004661' 201 01 0 00 004742'                 MOVEI   1,T25M                  ; set up microcode address
004662' 260 17 0 00 004467*                 GO      TLOAD                   ; load/verify it
004663' 263 17 0 00 000000                  RTN                             ; failed - exit test

                                    ; Initialization

004664' 400 15 0 00 000000          TL25:   SETZ    ERFLG,                  ; clear error flag
004665' 201 06 0 00 004702'                 MOVEI   6,TS25                  ; get test table address
004666' 402 00 0 00 004542*                 SETZM   TSTSUB                  ; initialize subtest number
004667' 476 00 0 00 004474*                 SETOM   TADDR                   ; clear start address
```

```

                                   ; Loop on test execute table entries

004670' 260 17 0 00 004475*        TA25:   GO      IPACLR                 ; clear port
004671' 260 17 0 00 004476*        TB25:   GO      TEXEC                  ; execute table entry
004672' 254 00 0 00 004701'                 JRST   TX25                   ; end of sstep table
004673' 254 00 0 00 004671'                 JRST   TB25                   ; keep looping after call
004674' 474 15 0 00 000000                  SETO   ERFLG,                 ; error occurred

                                   ; Handle error printouts and scope looping

004675' 027 00 0 00 004731'                SCOPER  MA25                   ; print error message
004676' 254 00 0 00 004664'                JRST    TL25                   ; loop on error
004677' 254 00 0 00 004701'                JRST    TX25                   ; altmode exit
004700' 322 15 0 00 004670'                JUMPE   ERFLG,TA25             ; do next sstep table entry

                                   ; End of test

004701' 263 17 0 00 000000         TX25:   RTN                            ; return

                                   ; Test Execute Table, as:  (CMD,parameters)

                                   ; CC is cleared initially

004702' 040000  000000             TS25:   TTABLE  (TSTART,0)             ; start up port
004703' 240002  000000                     TTABLE  (TCHECK,2)             ; check if completed
004704' 340000  000000                     TTABLE  (TEXIT)                ; exit if error yet

                                   ; CC sets when an interrupt is pending

004705' 140000  004722'                    TTABLE  (TCALL,TS25IA)         ; initialize (to not handle interrupts)
004706' 040000  000010                     TTABLE  (TSTART,10)            ; start up port
004707' 240002  000000                     TTABLE  (TCHECK,2)             ; check if completed
004710' 340000  000000                     TTABLE  (TEXIT)                ; exit if error yet
004711' 200000  004717'                    TTABLE  (TCALLC,TS25EX)        ; check if exit test

                                   ; CC clears after an interrupt has been handled

004712' 140000  004725'                    TTABLE  (TCALL,TS25IB)         ; initialize (to handle interrupts)
004713' 040000  000020                     TTABLE  (TSTART,20)            ; start up port
004714' 240010  000000                     TTABLE  (TCHECK,10)            ; check if completed
004715' 340000  000000                     TTABLE  (TEXIT)                ; exit if error yet
004716' 000000  000000                     TTABLE  (TLAST)                ; end of list

                                   ; Skip next test segment in user mode

004717' 336 00 0 00 030037         TS25EX: SKIPN   USER                   ; user mode?
004720' 350 00 0 17 000000                 AOS     (P)                    ; no - set up proper return
004721' 263 17 0 00 000000                 RTN                            ; return
```

```

                                          ; Set up start addresses

004722' 201 00 0 00 000010        TS25IA: MOVEI   10                        ; get initial start data
004723' 202 00 0 00 001051*               MOVEM   SDATA                     ; save it
004724' 263 17 0 00 000000                RTN

004725' 402 00 0 00 004723*       TS25IB: SETZM   SDATA                     ; set up initial start data
004726' 260 17 0 00 000000*               GO      INITPI                    ; initialize PI system
004727' 260 17 0 00 000000*               GO      SETVEC                    ; set up interrupt vectors ...
004730' 263 17 0 00 000000                RTN

                                  ; Error messages

004731' 140000  012305'           MA25:   MSG!TXNOT![ASCIZ /Condition Code 'CC Intr Active' not /]
004732' 240000  004734'                   CALL!TXNOT!MA25LP                 ; print subtest description
004733' 000000000000#                     LAST!CALL!TXALL!TTPNT             ; print test data

004734' 200 01 0 00 004666*       MA25LP: MOVE    1,TSTSUB                  ; get subtest number
004735' 256 00 0 01 004736'               XCT     MA25L-1(1)                ; print message
004736' 263 17 0 00 000000                RTN                               ; return

004737' 037 00 0 00 011656'       MA25L:  TMSG    <cleared initially>
004740' 037 00 0 00 012315'               TMSG    <set when an interrupt is pending>
                                          TMSG    <cleared after an interrupt has
004741' 037 00 0 00 012324'         been handled>

                                  ; Microcode:

                                  ;       CC is cleared initially

004742' 000000  010000            T25M:   MWORD   <ADDR=0,JMAP,J=1,SOA,AND,D=2>                           ; 0
004743' 442000  000040
004744' 000152  000000                    MWORD   <CJP,J=5200,CENA,CCIA>                                  ; 1
004745' 000400  110060
004746' 000251  000000                    MWORD   <JMAP,J=5100>                                           ; 2
004747' 000000  000040

                                  ;       CC sets when an interrupt is pending

004750' 001000  112004                    MWORD   <ADDR=10,JMAP,J=11,SOA,AND,D=2,OENA,SELE,MGC=4> ; 10
004751' 442000  005040
004752' 001100  000002                    MWORD   <CONT,SELE,MGC=2>                                       ; 11
004753' 000000  005340
004754' 001251  000000                    MWORD   <CJP,J=5100,CENA,CCIA>                                  ; 12
004755' 000400  110060
004756' 001352  000000                    MWORD   <JMAP,J=5200>                                           ; 13
004757' 000000  000040

                                  ;       CC clears after an interrupt has been handled

004760' 002000  212004                    MWORD   <ADDR=20,JMAP,J=21,SOA,AND,D=2,OENA,SELE,MGC=4> ; 20
004761' 442000  005040
004762' 002100  000002                    MWORD   <CONT,SELE,MGC=2>                                       ; 21
004763' 000000  005340
```

```
004764' 002200  220000            MWORD   <CJP,J=22,CENA,CCIA>                          ; 22
004765' 000400  110060
004766' 002351  000000            MWORD   <JMAP,J=5100>                                 ; 23
004767' 000000  000040

                         ; Subroutine - Passed - Write 1 to bit 12 of CSR register

004770' 510000  000000            MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>        ; 5100
004771' 742001  000340
004772' 510100  000000            MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5101
004773' 302001  000740
004774' 510200  260000            MWORD   <LDCT,J=26>                             ; 5102
004775' 000000  000300
004776' 510351  030000            MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>      ; 5103
004777' 102021  000220
005000' 510400  002004            MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5104
005001' 431020  005340
005002' 510551  070040            MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>  ; 5105
005003' 001400  015060
005004' 510651  050000            MWORD   <JMAP,J=5105,D=1>                       ; 5106
005005' 001000  000040
005006' 510700  000200            MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5107
005007' 431010  005340
005010' 511051  100000            MWORD   <JMAP,J=5110,D=1>                       ; 5110
005011' 001000  000040

                         ; Subroutine - Failed - Write 1 to bit 13 of CSR register

005012' 520000  000000            MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>        ; 5200
005013' 742001  000340
005014' 520100  000000            MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5201
005015' 302001  000740
005016' 520200  250000            MWORD   <LDCT,J=25>                             ; 5202
005017' 000000  000300
005020' 520352  030000            MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>      ; 5203
005021' 102021  000220
005022' 520400  002004            MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5204
005023' 431020  005340
005024' 520552  070040            MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>  ; 5205
005025' 001400  015060
005026' 520652  050000            MWORD   <JMAP,J=5205,D=1>                       ; 5206
005027' 001000  000040
005030' 520700  000200            MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5207
005031' 431010  005340
005032' 521052  100000            MWORD   <JMAP,J=5210,D=1>                       ; 5210
005033' 001000  000040
005034' 777777  777777            -1
```

```
                                        ;#*********************************************************************
                                        ;* TEST 26 - CCMBSign Test
                                        ;
                                        ; Description:  Verify that  condition  code  'CCMBSign'  can  be
                                        ;               tested  by  the  microsequencer  for  conditional
                                        ;               branching, and that the condition code can be set
                                        ;               and reset by the hardware conditions which should
                                        ;               do so.
                                        ;
                                        ; Procedure:    Port Clear
                                        ;               Start up port
                                        ;               UC> perform an operation and then verify that
                                        ;                   CCMBSign is not asserted or asserted
                                        ;               UC> if passed - set bit 12 (DISABLE Complete)
                                        ;               UC> if failed - set bit 13 (ENABLE Complete)
                                        ;               Read CSR and determine test disposition
                                        ;
                                        ;               Repeat with several test cases:
                                        ;                 CC is cleared when ALU output is zero
                                        ;                 CC is set when ALU output is -1
                                        ;                 CC is cleared when ALU output is 377777,,-1
                                        ;                 CC is set when ALU output is 400000,,0
                                        ;
                                        ; Failure:      ---
                                        ;#*********************************************************************

                                        ; Test data

005035' 254 00 0 00 005047'             TSTU26: JRST    TG26                            ; go start test
005036' 200403  000026                          MPROC!NDMP!ZMPROC!26                    ; test mask
005037' 005121' 012336'                         T26M,,[ASCIZ ^CCMBSign Test^]
005040' 012012' 012014'                         [EXP M6,MLAST!E23],,[EXP MLAST!M7]
005041' 000000  005216'                         TSTU27                                  ; failure test table
005042' 000000  005512'                         TSTU30                                  ;   ...
005043' 000000  005747'                         TSTU31
005044' 000000  006307'                         TSTU32
005045' 000000  006603'                         TSTU33
005046' 777777  777777                          -1

                                        ; Start test

005047' 201 00 0 00 000000'             TG26:   MOVEI   Z8                              ; get address of module start
005050' 260 17 0 00 004660*                     GO      TRACE                           ; handle trace output
005051' 201 01 0 00 005121'                     MOVEI   1,T26M                          ; set up microcode address
005052' 260 17 0 00 004662*                     GO      TLOAD                           ; load/verify it
005053' 263 17 0 00 000000                      RTN                                     ; failed - exit test
```

```
                                       ; Initialization

005054' 400 15 0 00 000000     TL26:   SETZ    ERFLG,                  ; clear error flag
005055' 201 06 0 00 005072'            MOVEI   6,TS26                  ; get test table address
005056' 402 00 0 00 004734*            SETZM   TSTSUB                  ; initialize subtest number
005057' 476 00 0 00 004667*            SETOM   TADDR                   ; clear start address

                                       ; Loop on test execute table entries

005060' 260 17 0 00 004670*    TA26:   GO      IPACLR                  ; clear port
005061' 260 17 0 00 004671*    TB26:   GO      TEXEC                   ; execute table entry
005062' 254 00 0 00 005071'             JRST   TX26                    ; end of sstep table
005063' 254 00 0 00 005061'             JRST   TB26                    ; keep looping after call
005064' 474 15 0 00 000000              SETO   ERFLG,                  ; error occurred

                                       ; Handle error printouts and scope looping

005065' 027 00 0 00 005107'            SCOPER  MA26                    ; print error message
005066' 254 00 0 00 005054'            JRST    TL26                    ; loop on error
005067' 254 00 0 00 005071'            JRST    TX26                    ; altmode exit
005070' 322 15 0 00 005060'            JUMPE   ERFLG,TA26              ; do next sstep table entry

                                       ; End of test

005071' 263 17 0 00 000000     TX26:   RTN                             ; return

                                       ; Test Execute Table, as:  (CMD,parameters)

                                       ; CC is cleared when ALU output is zero

005072' 040000  000000         TS26:   TTABLE  (TSTART,0)              ; start up port
005073' 240002  000000                 TTABLE  (TCHECK,2)              ; check if completed
005074' 340000  000000                 TTA3LE  (TEXIT)                 ; exit if error yet

                                       ; CC is set when ALU output is -1

005075' 040000  000010                 TTABLE  (TSTART,10)             ; start up port
005076' 240002  000000                 TTABLE  (TCHECK,2)              ; check if completed
005077' 340000  000000                 TTABLE  (TEXIT)                 ; exit if error yet

                                       ; CC is cleared when ALU output is 377777,,-1

005100' 040000  000020                 TTABLE  (TSTART,20)             ; start up port
005101' 240002  000000                 TTABLE  (TCHECK,2)              ; check if completed
005102' 340000  000000                 TTABLE  (TEXIT)                 ; exit if error yet
```

```

                                        ; CC is set when ALU output is 400000,,0

005103' 040000  000030                          TTABLE  (TSTART,30)             ; start up port
005104' 240002  000000                          TTABLE  (TCHECK,2)              ; check if completed
005105' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet
005106' 000000  000000                          TTABLE  (TLAST)                 ; end of list

                                        ; Error messages

005107' 140000  012341'                 MA26:   MSG!TXNOT![ASCIZ /Condition Code 'CC MB Sign' not /]
005110' 240000  005112'                         CALL!TXNOT!MA26LP               ; print subtest description
005111' 000000000000#                           LAST!CALL!TXALL!TTPNT           ; print test data

005112' 200 01 0 00 005056*             MA26LP: MOVE    1,TSTSUB                ; get subtest number
005113' 256 00 0 01 005114'                     XCT     MA26L-1(1)              ; print message
005114' 263 17 0 00 000000                      RTN                             ; return

005115' 037 00 0 00 012350'             MA26L:  TMSG    <cleared when ALU output is zero>
005116' 037 00 0 00 012357'                     TMSG    <set when ALU output is -1>
005117' 037 00 0 00 012365'                     TMSG    <cleared when ALU output is 377777,,-1>
005120' 037 00 0 00 012375'                     TMSG    <set when ALU output is 400000,,0>

                                        ; Microcode:

                                        ;       CC is cleared when ALU output is zero

005121' 000000  010000                  T26M:   MWORD   <ADDR=0,JMAP,J=1,SOA,AND,D=1>           ; 0
005122' 441000  000040
005123' 000152  000000                          MWORD   <CJP,J=5200,SOA,AND,D=1,CENA,CCMB>      ; 1
005124' 441400  120060
005125' 000251  000000                          MWORD   <JMAP,J=5100>                           ; 2
005126' 000000  000040

                                        ;       CC is set when ALU output is -1

005127' 001000  110000                          MWORD   <ADDR=10,JMAP,J=11,SOA,AND,D=1>         ; 10
005130' 441000  000040
005131' 001151  000000                          MWORD   <CJP,J=5100,SAB,XNOR,D=1,CENA,CCMB>     ; 11
005132' 171400  120060
005133' 001252  000000                          MWORD   <JMAP,J=5200>                           ; 12
005134' 000000  000040

                                        ;       CC is cleared when ALU output 377777,,-1

005135' 002000  210777                          MWORD   <ADDR=20,JMAP,J=21,SDO,OR,D=2,MGC=777,SKCN>
005136' 732000  240040
005137' 002152  000000                          MWORD   <CJP,J=5200,SOA,OR,D=1,CENA,CCMB>       ; 21
005140' 431400  120060
005141' 002251  000000                          MWORD   <JMAP,J=5100>                           ; 22
005142' 000000  000040

                                        ;       CC is set when ALU output 400000,,0

005143' 003000  311000                          MWORD   <ADDR=30,JMAP,J=31,SDO,OR,D=2,MGC=1000,SKCN>
```

```
 4263  005144' 732000  240040            MWORD   <CJP,J=5100,SOA,OR,D=1,CENA,CCMB>        ; 31
 4264  005145' 003151  000000
 4265  005146' 431400  120060            MWORD   <JMAP,J=5200>                            ; 32
 4266  005147' 003252  000000
 4267  005150' 000000  000040
 4268
 4269                            ; Subroutine - Passed - Write 1 to bit 12 of CSR register
 4270
 4271  005151' 510000  000000            MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>         ; 5100
 4272  005152' 742001  000340
 4273  005153' 510100  000000            MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>              ; 5101
 4274  005154' 302001  000740
 4275  005155' 510200  260000            MWORD   <LDCT,J=26>                              ; 5102
 4276  005156' 000000  000300
 4277  005157' 510351  030000            MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>       ; 5103
 4278  005160' 102021  000220
 4279  005161' 510400  002004            MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>    ; 5104
 4280  005162' 431020  005340
 4281  005163' 510551  070040            MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>   ; 5105
 4282  005164' 001400  015060
 4283  005165' 510651  050000            MWORD   <JMAP,J=5105,D=1>                        ; 5106
 4284  005166' 001000  000040
 4285  005167' 510700  000200            MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>       ; 5107
 4286  005170' 431010  005340
 4287  005171' 511051  100000            MWORD   <JMAP,J=5110,D=1>                        ; 5110
 4288  005172' 001000  000040
 4289
 4290                            ; Subroutine - Failed - Write 1 to bit 13 of CSR register
 4291
 4292  005173' 520000  000000            MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>         ; 5200
 4293  005174' 742001  000340
 4294  005175' 520100  000000            MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>              ; 5201
 4295  005176' 302001  000740
 4296  005177' 520200  250000            MWORD   <LDCT,J=25>                              ; 5202
 4297  005200' 000000  000300
 4298  005201' 520352  030000            MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>       ; 5203
 4299  005202' 102021  000220
 4300  005203' 520400  002004            MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>    ; 5204
 4301  005204' 431020  005340
 4302  005205' 520552  070040            MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>   ; 5205
 4303  005206' 001400  015060
 4304  005207' 520652  050000            MWORD   <JMAP,J=5205,D=1>                        ; 5206
 4305  005210' 001000  000040
 4306  005211' 520700  000200            MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>       ; 5207
 4307  005212' 431010  005340
 4308  005213' 521052  100000            MWORD   <JMAP,J=5210,D=1>                        ; 5210
 4309  005214' 001000  000040
 4310  005215' 777777  777777            -1
```

```

                                        ;#**********************************************************************
                                        ;* TEST 27 - CCMVParChk Test
                                        ;
                                        ; Description:  Verify that  condition  code  'CCMVParChk' can be
                                        ;               tested  by  the  microsequencer  for  conditional
                                        ;               branching, and that the condition code can be set
                                        ;               and reset by the hardware conditions which should
                                        ;               do so.
                                        ;
                                        ; Procedure:    Port Clear
                                        ;               Start up port
                                        ;               UC> perform an operation and then verify that
                                        ;                       CCMVParChk is not asserted or asserted
                                        ;               UC> if passed - set bit 12 (DISABLE Complete)
                                        ;               UC> if failed - set bit 13 (ENABLE Complete)
                                        ;               Read CSR and determine test disposition
                                        ;
                                        ;               Repeat with several test cases:
                                        ;                 CC is initially cleared
                                        ;                 CC is set after reading a word with parity bit
                                        ;                   set from the CBUS
                                        ;                 CC clears after reading a word with parity bit
                                        ;                   not set from the CBUS
                                        ;                 CC does not clear by setting 'MP Clr CC Code'
                                        ;                 CC clears by setting 'MP Clr Par Chk'
                                        ;
                                        ; Failure:      ---
                                        ;#**********************************************************************

                                        ; Test data

005216' 254 00 0 00 005230'             TSTU27: JRST    TG27                    ; go start test
005217' 200403  000027                          MPROC!NDMP!ZMPROC!27            ; test mask
005220' 005333' 012404'                         T27M,,[ASCIZ ^CCMVParChk Test^]
005221' 012410' 012412'                         [EXP M6,MLAST!C11],,[EXP M7,C16,MLAST!C8]
005222' 000000000000#                           TSTC3                           ; failure test table
005223' 000000000000#                           TSTC4                           ;   ...
005224' 000000  005512'                         TSTU30
005225' 000000  006307'                         TSTU32
005226' 000000  006603'                         TSTU33
005227' 777777  777777                          -1

                                        ; Start test

005230' 201 00 0 00 000000'             TG27:   MOVEI   Z8                      ; get address of module start
005231' 260 17 0 00 005050*                     GO      TRACE                   ; handle trace output
005232' 201 01 0 00 005333'                     MOVEI   1,T27M                  ; set up microcode address
005233' 260 17 0 00 005052*                     GO      TLOAD                   ; load/verify it
005234' 263 17 0 00 000000                      RTN                             ; failed - exit test
```

```

                                        ; Initialization

005235' 400 15 0 00 000000      TL27:   SETZ    ERFLG,                  ; clear error flag
005236' 260 17 0 00 005060*             GO      IPACLR                  ; clear the CSR register
005237' 402 00 0 00 005112*             SETZM   TSTSUB                  ; initialize subtest number
005240' 201 06 0 00 005261'             MOVEI   6,TS27                  ; get test table address
005241' 476 00 0 00 005057*             SETOM   TADDR                   ; clear start address
005242' 201 01 0 00 005000              MOVEI   1,5000                  ; get start address
005243' 242 01 0 00 000001              LSH     1,1                     ; position correctly
005244' 260 17 0 00 004535*             GO      LDRAR                   ; load the register
005245' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get 'MPRUN' bit
005246' 260 17 0 00 003572*             GO      LDCSR                   ; start up the port

                                        ; Loop on test execute table entries

005247' 260 17 0 00 005236*     TA27:   GO      IPACLR                  ; clear port
005250' 260 17 0 00 005061*     TB27:   GO      TEXEC                   ; execute table entry
005251' 254 00 0 00 005260'              JRST   TX27                    ; end of sstep table
005252' 254 00 0 00 005250'              JRST   TB27                    ; keep looping after call
005253' 474 15 0 00 000000               SETO   ERFLG,                  ; error occurred

                                        ; Handle error printouts and scope looping

005254' 027 00 0 00 005320'             SCOPER  MA27                    ; print error message
005255' 254 00 0 00 005235'             JRST    TL27                    ; loop on error
005256' 254 00 0 00 005260'             JRST    TX27                    ; altmode exit
005257' 322 15 0 00 005250'             JUMPE   ERFLG,TB27              ; do next sstep table entry

                                        ; End of test

005260' 263 17 0 00 000000      TX27:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

                                        ; CC is cleared initially

005261' 040000  000000          TS27:   TTABLE  (TSTART,0)              ; start up port
005262' 240010  000000                  TTABLE  (TCHECK,10)             ; check if completed
005263' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; CC is set after reading a word with parity bit
                                        ; set from the CBUS

005264' 140000  005305'                 TTABLE  (TCALL,T27SR1)          ; set up read of 1 word of 000..01
005265' 040000  000010                  TTABLE  (TSTART,10)             ; start up port
005266' 240010  000000                  TTABLE  (TCHECK,10)             ; check if completed
005267' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

```

```

                                        ; CC clears after reading a word with parity bit
                                        ; not set from the CBUS

005270' 140000  005307'                         TTABLE  (TCALL,T27SR2)          ; set up read of 1 word of 000..00
005271' 040000  000020                          TTABLE  (TSTART,20)             ; start up port
005272' 240010  000000                          TTABLE  (TCHECK,10)             ; check if completed
005273' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; CC does not clear by setting 'MP Clr CC Code'

005274' 140000  005305'                         TTABLE  (TCALL,T27SR1)          ; set up read of 1 word of 000..01
005275' 040000  000030                          TTABLE  (TSTART,30)             ; start up port
005276' 240010  000000                          TTABLE  (TCHECK,10)             ; check if completed
005277' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; CC clears by setting 'MP Clr Par Chk'

005300' 140000  005305'                         TTABLE  (TCALL,T27SR1)          ; set up read of 1 word of 000..01
005301' 040000  000040                          TTABLE  (TSTART,40)             ; start up port
005302' 240010  000000                          TTABLE  (TCHECK,10)             ; check if completed
005303' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet
005304' 000000  000000                          TTABLE  (TLAST)                 ; end of list

                                        ; Set up data transfer

005305' 201 00 0 00 011111              T27SR1: MOVEI   11111                   ; get word with odd parity
005306' 334 00 0 00 000000                      SKIPA
005307' 201 00 0 00 002222              T27SR2: MOVEI   2222                    ; get word with even parity
005310' 202 00 0 00 002037*                     MOVEM   BUFF                    ; save in buffer
005311' 200 01 0 00 011645'                     MOVE    1,[-BUFSZ,,CBUF]        ; get pointer to control buffer
005312' 260 17 0 00 002036*                     GO      [illegible]             ; initialize software
005313' 551 01 0 00 005310*                     HRRZI   1,BUFF                  ; buffer address
005314' 201 02 0 00 000001                      MOVEI   2,1                     ; word count
005315' 400 03 0 00 000000                      SETZ    3,                      ; clear zero fill specifier
005316' 260 17 0 00 002042*                     GO      GENCCW                  ; generate a CCW list
005317' 263 17 0 00 000000                      RTN                             ; return

                                        ; Error messages

005320' 140000  012415'                 MA27:   MSG!TXNOT![ASCIZ /Condition Code 'CC MV Par Chk' not /]
005321' 240000  005323'                         CALL!TXNOT!MA27LP               ; print subtest description
005322' 000000000000#                           LAST!CALL!TXALL!TTPNT           ; print test data

005323' 200 01 0 00 005237*             MA27LP: MOVE    1,TSTSUB                ; get subtest number
005324' 256 00 0 01 005325'                     XCT     MA27L-1(1)              ; print message
005325' 263 17 0 00 000000                      RTN                             ; return
```

```
005326' 037 00 0 00 012425'     MA27L:  TMSG    <initially cleared>
                                        TMSG    <set after reading a word with
005327' 037 00 0 00 012431'       parity bit set from the CBUS>
                                        TMSG    <cleared after reading a word
005330' 037 00 0 00 012446'       with parity bit not set from the CBUS>
                                        TMSG    <still set after setting
005331' 037 00 0 00 012464'       'MP Clr CC Code'>
                                        TMSG    <cleared by setting 'MP Clr
005332' 037 00 0 00 012475'       Par Chk'>

                                ; Microcode:

                                ;       CC is cleared initially

005333' 000000  010000          T27M:   MWORD   <ADDR=0,JMAP,J=1>                ; 0
005334' 000000  000040
005335' 000152  000000                  MWORD   <CJP,J=5200,CENA,CCMP>           ; 1
005336' 000400  130060
005337' 000251  000000                  MWORD   <JMAP,J=5100>                    ; 2
005340' 000000  000040

                                ; CC is set after reading a word with parity bit
                                ; set from the CBUS

005341' 001000  110000                  MWORD   <ADDR=10,JMAP,J=11>              ; 10
005342' 000000  000040
005343' 001120  000000                  MWORD   <CJS,J=2000>                     ; 11
005344' 000000  000020
005345' 001251  000000                  MWORD   <CJP,J=5100,CENA,CCMP>           ; 12
005346' 000400  130060
005347' 001352  000000                  MWORD   <JMAP,J=5200>                    ; 13
005350' 000000  000040

                                ; CC clears after reading a word with parity bit
                                ; not set from the CBUS

005351' 002000  210000                  MWORD   <ADDR=20,JMAP,J=21>              ; 20
005352' 000000  000040
005353' 002120  000000                  MWORD   <CJS,J=2000>                     ; 21
005354' 000000  000020
005355' 002252  000000                  MWORD   <CJP,J=5200,CENA,CCMP>           ; 22
005356' 000400  130060
005357' 002351  000000                  MWORD   <JMAP,J=5100>                    ; 23
005360' 000000  000040

                                ; CC does not clear by setting 'MP Clr CC Code'

005361' 003000  310000                  MWORD   <ADDR=30,JMAP,J=31>              ; 30
005362' 000000  000040
005363' 003120  000000                  MWORD   <CJS,J=2000>                     ; 31
005364' 000000  000020
005365' 003200  000020                  MWORD   <CONT,SELM,MGC=20>               ; 32
005366' 000000  002340
005367' 003351  000000                  MWORD   <CJP,J=5100,CENA,CCMP>           ; 33
```

```
005370' 000400  130060
005371' 003452  000000                  MWORD   <JMAP,J=5200>                           ; 34
005372' 000000  000040

                                ; CC clears by setting 'MP Clr Par Chk'

005373' 004000  410000                  MWORD   <ADDR=40,JMAP,J=41>                     ; 40
005374' 000000  000040
005375' 004120  000000                  MWORD   <CJS,J=2000>                            ; 41
005376' 000000  000020
005377' 004200  000010                  MWORD   <CONT,SELM,MGC=10>                      ; 42
005400' 000000  002340
005401' 004352  000000                  MWORD   <CJP,J=5200,CENA,CCMP>                  ; 43
005402' 000400  130060
005403' 004451  000000                  MWORD   <JMAP,J=5100>                           ; 44
005404' 000000  000040

                                ;       Initialization step

005405' 500050  010100                  MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>    ; 5000
005406' 000000  004040
005407' 500150  010020                  MWORD   <JMAP,J=5001,SELM,MGC=20>               ; 5001
005410' 000000  002040
005411' 500250  020000                  MWORD   <JMAP,J=5002>                           ; 5002
005412' 000000  000040

                                ; Subroutine to read a word over the CBUS

005413' 200020  010020                  MWORD   <ADDR=2000,JMAP,J=2001,SDO,AND,D=2,SELM,MGC=20>
005414' 742000  002040
005415' 200100  000100                  MWORD   <CONT,D=1,SELC,MGC=100>                 ; 2001 (stop CBUS)
005416' 001000  004340
005417' 200201  000000                  MWORD   <LDCT,J=100,D=1>                        ; 2002
005420' 001000  000300
005421' 200320  030000                  MWORD   <RPCT,J=2003,D=1>                       ; 2003
005422' 001000  000220
005423' 200400  000200                  MWORD   <CONT,D=1,SELC,MGC=200>                 ; 2004 (start CBUS)
005424' 001000  004340
005425' 200500  100000                  MWORD   <LDCT,J=10,D=1>                         ; 2005
005426' 001000  000300
005427' 200620  060000                  MWORD   <RPCT,J=2006,D=1>                       ; 2006
005430' 001000  000220
005431' 200720  110000                  MWORD   <CJP,J=2011,D=1,CENA,CCCA>              ; 2007 (wait til avail)
005432' 001400  000060
005433' 201020  070000                  MWORD   <JMAP,J=2007,D=1>                       ; 2010
005434' 001000  000040
005435' 201100  000004                  MWORD   <CONT,D=1,SELC,MGC=4>                   ; 2011
005436' 001000  004340
005437' 201200  000200                  MWORD   <CONT,SDO,OR,D=2,SELM,MGC=200>          ; 2012
005440' 732000  002340
005441' 201300  002004                  MWORD   <CONT,SOA,OR,D=1,OENA,SELE,MGC=4>       ; 2013
005442' 431000  005340
005443' 201400  000140                  MWORD   <CRTN,SELC,MGC=140>                     ; 2014 (STOP,STORE)
005444' 000000  004240
```

```
                                ; Subroutine - Passed - Write 1 to bit 12 of CSR register

005445' 510000  000000                  MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>            ; 5100
005446' 742001  000340
005447' 510100  000000                  MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                 ; 5101
005450' 302001  000740
005451' 510200  260000                  MWORD   <LDCT,J=26>                                 ; 5102
005452' 000000  000300
005453' 510351  030000                  MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>          ; 5103
005454' 102021  000220
005455' 510400  002004                  MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>       ; 5104
005456' 431020  005340
005457' 510551  070040                  MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>      ; 5105
005460' 001400  015060
005461' 510651  050000                  MWORD   <JMAP,J=5105,D=1>                           ; 5106
005462' 001000  000040
005463' 510700  000200                  MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>          ; 5107
005464' 431010  005340
005465' 511051  100000                  MWORD   <JMAP,J=5110,D=1>                           ; 5110
005466' 001000  000040

                                ; Subroutine - Failed - Write 1 to bit 13 of CSR register

005467' 520000  000000                  MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>            ; 5200
005470' 742001  000340
005471' 520100  000000                  MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                 ; 5201
005472' 302001  000740
005473' 520200  250000                  MWORD   <LDCT,J=25>                                 ; 5202
005474' 000000  000300
005475' 520352  030000                  MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>          ; 5203
005476' 102021  000220
005477' 520400  002004                  MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>       ; 5204
005500' 431020  005340
005501' 520552  070040                  MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>      ; 5205
005502' 001400  015060
005503' 520652  050000                  MWORD   <JMAP,J=5205,D=1>                           ; 5206
005504' 001000  000040
005505' 520700  000200                  MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>          ; 5207
005506' 431010  005340
005507' 521052  100000                  MWORD   <JMAP,J=5210,D=1>                           ; 5210
005510' 001000  000040
005511' 777777  777777                  -1
```

```

                                         ;#*****************************************************************
                                         ;* TEST 30 - CCCbusParErr Test
                                         ;
                                         ; Description:  Verify that  condition code 'CCCbusParErr' can be
                                         ;               tested  by  the  microsequencer  for  conditional
                                         ;               branching, and that the condition code can be set
                                         ;               and reset by the hardware conditions which should
                                         ;               do so.
                                         ;
                                         ; Procedure:    Port Clear
                                         ;               Start up port
                                         ;               UC> perform an operation and then verify that
                                         ;                   CCCbusParErr is not asserted or asserted
                                         ;               UC> if passed - set bit 12 (DISABLE Complete)
                                         ;               UC> if failed - set bit 13 (ENABLE Complete)
                                         ;               Read CSR and determine test disposition
                                         ;
                                         ;               Repeat with several test cases:
                                         ;                 CC is initially cleared
                                         ;                 CC is cleared after reading each word of a
                                         ;                   floating 1's and 0's pattern
                                         ;                 CC is cleared after a CLRCCCODE command
                                         ;
                                         ; Failure:      ---
                                         ;#***  ************************************************************

                                         ; Test data

 005512' 254 00 0 00 005523'             TSTU30: JRST    TG30                          ; go start test
 005513' 200403  000030                          MPROC!NDMP!ZMPROC!30                  ; test mask
 005514' 005640' 012505'                         T30M,,[ASCIZ ^CCCbusParErr Test^]
 005515' 012511' 012513'                         [EXP M6,MLAST!C9],,[EXP M7,C17,C12,MLAST!C7]
 005516' 000000000000#                           TSTC3                                 ; failure test table
 005517' 000000000000#                           TSTC4                                 ;   ...
 005520' 000000  006307'                         TSTU32
 005521' 000000  006603'                         TSTU33
 005522' 777777  777777                          -1

                                         ; Start test

 005523' 201 00 0 00 000000'             TG30:   MOVEI   Z8                            ; get address of module start
 005524' 260 17 0 00 005231*                     GO      TRACE                         ; handle trace output
 005525' 201 01 0 00 005640'                     MOVEI   1,T30M                        ; set up microcode address
 005526' 260 17 0 00 005233*                     GO      TLOAD                         ; load/verify it
 005527' 263 17 0 00 000000                      RTN                                   ; failed - exit test

                                         ; Initialization

 005530' 400 15 0 00 000000              TL30:   SETZ    ERFLG,                        ; clear error flag
 005531' 201 06 0 00 005546'                     MOVEI   6,TS30                        ; get test table address
 005532' 402 00 0 00 005323*                     SETZM   TSTSUB                        ; initialize subtest number
 005533' 476 00 0 00 005241*                     SETOM   TADDR                         ; clear start address
```

```

                                        ; Loop on test execute table entries

005534' 260 17 0 00 005247*             TA30:   GO      IPACLR                  ; clear port
005535' 260 17 0 00 005250*             TB30:   GO      TEXEC                   ; execute table entry
005536' 254 00 0 00 005545'                      JRST   TX30                    ; end of sstep table
005537' 254 00 0 00 005535'                      JRST   TB30                    ; keep looping after call
005540' 474 15 0 00 000000                       SETO   ERFLG,                  ; error occurred

                                        ; Handle error printouts and scope looping

005541' 027 00 0 00 005600'                     SCOPER  MA30                    ; print error message
005542' 254 00 0 00 005530'                     JRST    TL30                    ; loop on error
005543' 254 00 0 00 005545'                     JRST    TX30                    ; altmode exit
005544' 322 15 0 00 005534'                     JUMPE   ERFLG,TA30              ; do next sstep table entry

                                        ; End of test

005545' 263 17 0 00 000000              TX30:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

                                        ; CC is cleared initially

005546' 040000  000000                  TS30:   TTABLE  (TSTART,0)              ; start up port
005547' 240010  000000                          TTABLE  (TCHECK,10)             ; check if completed
005550' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; CC is cleared after reading each word of a floating 1's and 0's pattern

005551' 140000  005560'                         TTABLE  (TCALL,T30SRD)          ; set up data transfer
005552' 040000  000010                          TTABLE  (TSTART,10)             ; start up port
005553' 240010  000000                          TTABLE  (TCHECK,10)             ; check if completed
005554' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; CC is cleared after a CLRCCCODE command

005555' 040000  000030                          TTABLE  (TSTART,30)             ; start up port
005556' 240010  000000                          TTABLE  (TCHECK,10)             ; check if completed
005557' 000000  000000                          TTABLE  (TLAST)                 ; end of list

```

```

                                    ; Set up data transfer

005560' 201 01 0 00 000001          T30SRD: MOVEI   1,1                     ; get initial floating 1
005561' 400 02 0 00 000000                  SETZ    2,                      ; get initial buffer offset
005562' 200 00 0 00 000001          T30SR1: MOVE    1                       ; get data pattern
005563' 450 00 0 00 000000                  SETCA                           ; complement it
005564' 202 01 0 02 005313*                 MOVEM   1,BUFF(2)               ; save floating 1
005565' 202 00 0 02 000000#                 MOVEM   BUFF+^D36(2)            ; save floating 0
005566' 350 00 0 00 000002                  AOS     2                       ; increment buffer offset
005567' 242 01 0 00 000001                  LSH     1,1                     ; generate new pattern
005570' 326 01 0 00 005562'                 JUMPN   1,T30SR1                ; loop till done
005571' 200 01 0 00 011645'                 MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
005572' 260 17 0 00 005312*                 GO      CHINIT                  ; initialize software
005573' 551 01 0 00 005564*                 HRRZI   1,BUFF                  ; buffer address
005574' 201 02 0 00 000110                  MOVEI   2,^D72                  ; word count
005575' 400 03 0 00 000000                  SETZ    3,                      ; clear zero fill specifier
005576' 260 17 0 00 005316*                 GO      GENCCW                  ; generate a CCW list
005577' 263 17 0 00 000000                  RTN                             ; return

                                    ; Error messages

005600' 140000  012517'             MA30:   MSG!TXNOT![ASCIZ /Condition Code 'CC Cbus Par Err' not /]
005601' 240000  005603'                     CALL!TXNOT!MA30LP               ; print subtest description
005602' 000000000000#                       LAST!CALL!TXALL!TTPNT           ; print test data

005603' 200 01 0 00 005532*         MA30LP: MOVE    1,TSTSUB                ; get subtest number
005604' 256 00 0 01 005605'                 XCT     MA30L-1(1)              ; print message
005605' 263 17 0 00 000000                  RTN                             ; return

005606' 037 00 0 00 012425'         MA30L:  TMSG    <initially cleared>
                                            TMSG    <cleared after reading each word
005607' 037 00 0 00 012527'           of a floating 1's and 0's pattern>
005610' 037 00 0 00 012161'                 TMSG    <cleared after a CLRCCCODE command>

005611' 260 17 0 00 000000*         MA30PN: GO      CHDATA                  ; check logout data for errors
005612' 336 00 0 00 000001                  SKIPN   1                       ; any errors?
005613' 263 17 0 00 000000                  RTN                             ; no - return
005614' 037 00 0 00 012545'                 TMSGC   <Data transfer errors:  >
005615' 603 01 0 00 200000                  TLNE    1,(MPER)                ; mem parity error
005616' 037 00 0 00 012553'                 TMSG    <MPERR >
005617' 603 01 0 00 100000                  TLNE    1,(ADRPE)               ; address parity error
005620' 037 00 0 00 012555'                 TMSG    <ADRPE >
005621' 603 01 0 00 040000                  TLNE    1,(NOTWCO)              ; not word count zero
005622' 037 00 0 00 012557'                 TMSG    <NOTWCO >
005623' 603 01 0 00 020000                  TLNE    1,(NEXM)                ; non-existant memory
005624' 037 00 0 00 012561'                 TMSG    <NEXM >
005625' 603 01 0 00 000400                  TLNE    1,(LAXER)               ; last xfer error
005626' 037 00 0 00 012563'                 TMSG    <LAXER >
005627' 603 01 0 00 000200                  TLNE    1,(IPAER)               ; IPA20 error
005630' 037 00 0 00 012565'                 TMSG    <IPAER >
```

```
005631' 603 01 0 00 000100                  TLNE    1,(LGWC)                ; long word count
005632' 037 00 0 00 012567'                 TMSG    <LGWC >
005633' 603 01 0 00 000040                  TLNE    1,(SHWC)                ; short word count
005634' 037 00 0 00 012571'                 TMSG    <SHWC >
005635' 603 01 0 00 000020                  TLNE    1,(OVN)                 ; overrrun
005636' 037 00 0 00 012573'                 TMSG    <OVN >
005637' 263 17 0 00 000000                  RTN                             ; return

                                    ; Microcode:

                                    ;       CC is cleared initially

005640' 000000  010000              T30M:   MWORD   <ADDR=0,JMAP,J=1>                       ; 0
005641' 000000  000040
005642' 000152  000000                      MWORD   <CJP,J=5200,CENA,CCCP>                  ; 1
005643' 000400  140060
005644' 000251  000000                      MWORD   <JMAP,J=5100>                           ; 2
005645' 000000  000040

                                    ;       CC is cleared after reading each word of a
                                    ;       floating 1's and 0's pattern

005646' 001000  110000                      MWORD   <ADDR=10,JMAP,J=11,SDO,AND,D=2>         ; 10
005647' 742000  000040
005650' 001101  070200                      MWORD   <LDCT,J=107,D=1,SELC,MGC=200>           ; 11 (start CBUS)
005651' 001000  004300
005652' 001200  140000                      MWORD   <CJP,J=14,D=1,CENA,CCCA>                ; 12 (wait till avail)
005653' 001400  000060
005654' 001300  120000                      MWORD   <JMAP,J=12,D=1>                         ; 13
005655' 001000  000040
005656' 001400  210000                      MWORD   <CJP,J=21,CENA,CCCP>                    ; 14
005657' 000400  140060
005660' 001500  170000                      MWORD   <JMAP,J=17>                             ; 15
005661' 000000  000040
005662' 001600  120000                      MWORD   <RPCT,J=12>                             ; 16
005663' 000000  000220
005664' 001700  000140                      MWORD   <CONT,SELC,MGC=140>                     ; 17 (STOP,STORE)
005665' 000000  004340
005666' 002051  000000                      MWORD   <JMAP,J=5100>                           ; 20
005667' 000000  000040
005670' 002152  000140                      MWORD   <JMAP,J=5200,SELC,MGC=140>              ; 21 (failed)
005671' 000000  004040

                                    ;       CC is cleared after a CLRCCCODE command

005672' 003000  310000                      MWORD   <ADDR=30,JMAP,J=31>                     ; 30
005673' 000000  000040
005674' 003100  000020                      MWORD   <CONT,SELM,MGC=20>                      ; 31
005675' 000000  002340
005676' 003252  000000                      MWORD   <CJP,J=5200,CENA,CCCP>                  ; 32
005677' 000400  140060
005700' 003351  000000                      MWORD   <JMAP,J=5100>                           ; 33
005701' 000000  000040

                                    ; Subroutine - Passed - Write 1 to bit 12 of CSR register
```

```

005702' 510000  000000                  MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>          ; 5100
005703' 742001  000340
005704' 510100  000000                  MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>               ; 5101
005705' 302001  000740
005706' 510200  260000                  MWORD   <LDCT,J=26>                               ; 5102
005707' 000000  000300
005710' 510351  030000                  MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>        ; 5103
005711' 102021  000220
005712' 510400  002004                  MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>     ; 5104
005713' 431020  005340
005714' 510551  070040                  MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>    ; 5105
005715' 001400  015060
005716' 510651  050000                  MWORD   <JMAP,J=5105,D=1>                         ; 5106
005717' 001000  000040
005720' 510700  000200                  MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>        ; 5107
005721' 431010  005340
005722' 511051  100000                  MWORD   <JMAP,J=5110,D=1>                         ; 5110
005723' 001000  000040

                              ; Subroutine - Failed - Write 1 to bit 13 of CSR register

005724' 520000  000000                  MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>          ; 5200
005725' 742001  000340
005726' 520100  000000                  MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>               ; 5201
005727' 302001  000740
005730' 520200  250000                  MWORD   <LDCT,J=25>                               ; 5202
005731' 000000  000300
005732' 520352  030000                  MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>        ; 5203
005733' 102021  000220
005734' 520400  002004                  MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>     ; 5204
005735' 431020  005340
005736' 520552  070040                  MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>    ; 5205
005737' 001400  015060
005740' 520652  050000                  MWORD   <JMAP,J=5205,D=1>                         ; 5206
005741' 001000  000040
005742' 520700  000200                  MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>        ; 5207
005743' 431010  005340
005744' 521052  100000                  MWORD   <JMAP,J=5210,D=1>                         ; 5210
005745' 001000  000040
005746' 777777  777777                  -1
```

```

                                       ;#*******************************************************************
                                       ;* TEST 31 - CCPLiParErr Test
                                       ;
                                       ; Description:  Verify that  condition code 'CCPLiParErr'  can be
                                       ;               tested  by  the  microsequencer  for  conditional
                                       ;               branching, and that the condition code can be set
                                       ;               and reset by the hardware conditions which should
                                       ;               do so.
                                       ;
                                       ; Procedure:    Port Clear
                                       ;               Start up port
                                       ;               UC> perform an operation and then verify that
                                       ;                   CCPLiParErr is not asserted or asserted
                                       ;               UC> if passed - set bit 12 (DISABLE Complete)
                                       ;               UC> if failed - set bit 13 (ENABLE Complete)
                                       ;               Read CSR and determine test disposition
                                       ;
                                       ;               Repeat with several test cases:
                                       ;                 CC is initially cleared
                                       ;                 CC is cleared after passing floating 1's through
                                       ;                   PLI Input/Output buffers
                                       ;                 CC is set after passing floating 1's through
                                       ;                   PLI Input/Output buffers with 'MPTESTPLIPAR'
                                       ;                   asserted
                                       ;                 CC is cleared after a CLRCCCODE command
                                       ;
                                       ; Failure:      ---
                                       ;#*******************************************************************

                                       ; Test data

 005747' 254 00 0 00 005756'           TSTU31: JRST    TG31                    ; go start test
 005750' 200403  000031                        MPROC!NDMP!ZMPROC!31            ; test mask
 005751' 006140' 012574'                       T31M,,[ASCIZ ^CCPLiParErr Test^]
 005752' 012600' 012602'                       [EXP M6,MLAST!C10],,[EXP M7,C21,MLAST!C8]
 005753' 000000  006307'                       TSTU32                          ; failure test table
 005754' 000000  006603'                       TSTU33                          ;   ...
 005755' 777777  777777                        -1

                                       ; Start test

 005756' 201 00 0 00 000000'           TG31:   MOVEI   Z8                      ; get address of module start
 005757' 260 17 0 00 005524*                   GO      TRACE                   ; handle trace output
 005760' 201 01 0 00 006140'                   MOVEI   1,T31M                  ; set up microcode address
 005761' 260 17 0 00 005526*                   GO      TLOAD                   ; load/verify it
 005762' 263 17 0 00 000000                    RTN                             ; failed - exit test
```

```

                                    ; Initialization

005763' 400 15 0 00 000000          TL31:   SETZ    ERFLG,                          ; clear error flag
005764' 201 06 0 00 006001'                 MOVEI   6,TS31                          ; get test table address
005765' 402 00 0 00 005603*                 SETZM   TSTSUB                          ; initialize subtest number
005766' 476 00 0 00 005533*                 SETOM   TADDR                           ; clear start address

                                    ; Loop on test execute table entries

005767' 260 17 0 00 005534*         TA31:   GO      IPACLR                          ; clear port
005770' 260 17 0 00 005535*         TB31:   GO      TEXEC                           ; execute table entry
005771' 254 00 0 00 006000'                  JRST   TX31                            ; end of sstep table
005772' 254 00 0 00 005770'                  JRST   TB31                            ; keep looping after call
005773' 474 15 0 00 000000                   SETO   ERFLG,                          ; error occurred

                                    ; Handle error printouts and scope looping

005774' 027 00 0 00 006070'                 SCOPER  MA31                            ; print error message
005775' 254 00 0 00 005763'                 JRST    TL31                            ; loop on error
005776' 254 00 0 00 006000'                 JRST    TX31                            ; altmode exit
005777' 322 15 0 00 005767'                 JUMPE   ERFLG,TA31                      ; do next sstep table entry

                                    ; End of test

006000' 263 17 0 00 000000          TX31:   RTN                                     ; return

                                    ; Test Execute Table, as:  (CMD,parameters)

                                    ; CC is cleared initially

006001' 040000  000000              TS31:   TTABLE  (TSTART,0)                      ; start up port
006002' 240002  000000                      TTABLE  (TCHECK,2)                      ; check if completed
006003' 340000  000000                      TTABLE  (TEXIT)                         ; exit if error yet

                                    ; CC is cleared after passing floating 1's through
                                    ;   PLI Input/Output buffers

006004' 040000  000010                      TTABLE  (TSTART,10)                     ; start up port
006005' 240002  000000                      TTABLE  (TCHECK,2)                      ; check if completed
006006' 340000  000000                      TTABLE  (TEXIT)                         ; exit if error yet
006007' 040000  000011                      TTABLE  (TSTART,11)                     ; start up port
006010' 240002  000000                      TTABLE  (TCHECK,2)                      ; check if completed
006011' 340000  000000                      TTABLE  (TEXIT)                         ; exit if error yet
006012' 040000  000012                      TTABLE  (TSTART,12)                     ; start up port
006013' 240002  000000                      TTABLE  (TCHECK,2)                      ; check if completed
006014' 340000  000000                      TTABLE  (TEXIT)                         ; exit if error yet
006015' 040000  000013                      TTABLE  (TSTART,13)                     ; start up port
006016' 240002  000000                      TTABLE  (TCHECK,2)                      ; check if completed
006017' 340000  000000                      TTABLE  (TEXIT)                         ; exit if error yet
006020' 040000  000014                      TTABLE  (TSTART,14)                     ; start up port
006021' 240002  000000                      TTABLE  (TCHECK,2)                      ; check if completed
006022' 340000  000000                      TTABLE  (TEXIT)                         ; exit if error yet

```

```
006023' 040000  000015              TTABLE  (TSTART,15)             ; start up port
006024' 240002  000000              TTABLE  (TCHECK,2)              ; check if completed
006025' 340000  000000              TTABLE  (TEXIT)                 ; exit if error yet
006026' 040000  000016              TTABLE  (TSTART,16)             ; start up port
006027' 240002  000000              TTABLE  (TCHECK,2)              ; check if completed
006030' 340000  000000              TTABLE  (TEXIT)                 ; exit if error yet
006031' 040000  000017              TTABLE  (TSTART,17)             ; start up port
006032' 240002  000000              TTABLE  (TCHECK,2)              ; check if completed
006033' 340000  000000              TTABLE  (TEXIT)                 ; exit if error yet

                            ; CC is set after passing floating 1's through
                            ;   PLI Input/Output buffers with 'MPTESTPLIPAR'
                            ;   asserted

006034' 040000  000030              TTABLE  (TSTART,30)             ; start up port
006035' 240002  000000              TTABLE  (TCHECK,2)              ; check if completed
006036' 340000  000000              TTABLE  (TEXIT)                 ; exit if error yet
006037' 040000  000031              TTABLE  (TSTART,31)             ; start up port
006040' 240002  000000              TTABLE  (TCHECK,2)              ; check if completed
006041' 340000  000000              TTABLE  (TEXIT)                 ; exit if error yet
006042' 040000  000032              TTABLE  (TSTART,32)             ; start up port
006043' 240002  000000              TTABLE  (TCHECK,2)              ; check if completed
006044' 340000  000000              TTABLE  (TEXIT)                 ; exit if error yet
006045' 040000  000033              TTABLE  (TSTART,33)             ; start up port
006046' 240002  000000              TTABLE  (TCHECK,2)              ; check if completed
006047' 340000  000000              TTABLE  (TEXIT)                 ; exit if error yet
006050' 040000  000034              TTABLE  (TSTART,34)             ; start up port
006051' 240002  000000              TTABLE  (TCHECK,2)              ; check if completed
006052' 340000  000000              TTABLE  (TEXIT)                 ; exit if error yet
006053' 040000  000035              TTABLE  (TSTART,35)             ; start up port
006054' 240002  000000              TTABLE  (TCHECK,2)              ; check if completed
006055' 340000  000000              TTABLE  (TEXIT)                 ; exit if error yet
006056' 040000  000036              TTABLE  (TSTART,36)             ; start up port
006057' 240002  000000              TTABLE  (TCHECK,2)              ; check if completed
006060' 340000  000000              TTABLE  (TEXIT)                 ; exit if error yet
006061' 040000  000037              TTABLE  (TSTART,37)             ; start up port
006062' 240002  000000              TTABLE  (TCHECK,2)              ; check if completed
006063' 340000  000000              TTABLE  (TEXIT)                 ; exit if error yet

                            ; CC is cleared after a CLRCCCODE command

006064' 040000  000050              TTABLE  (TSTART,50)             ; start up port
006065' 240002  000000              TTABLE  (TCHECK,2)              ; check if completed
006066' 340000  000000              TTABLE  (TEXIT)                 ; exit if error yet
006067' 000000  000000              TTABLE  (TLAST)                 ; end of list

                            ; Error messages

006070' 140000  012605'     MA31:   MSG!TXNOT![ASCIZ /Condition Code 'CC Pli Par Err' not /]
006071' 240000  006073'             CALL!TXNOT!MA31LP               ; print subtest description
006072' 000000000000#               LAST!CALL!TXALL!TTPNT           ; print test data
```

```
006073' 200 01 0 00 005765*     MA31LP: MOVE    1,TSTSUB                ; get subtest number
006074' 256 00 0 01 006075'             XCT     MA31L-1(1)              ; print message
006075' 263 17 0 00 000000              RTN                             ; return

006076' 037 00 0 00 012425'     MA31L:  TMSG    <initially cleared>
006077' 260 17 0 00 006120'             GO      MA31L1                  ; 1
006100' 260 17 0 00 006120'             GO      MA31L1                  ; 2
006101' 260 17 0 00 006120'             GO      MA31L1                  ; 4
006102' 260 17 0 00 006120'             GO      MA31L1                  ; 10
006103' 260 17 0 00 006120'             GO      MA31L1                  ; 20
006104' 260 17 0 00 006120'             GO      MA31L1                  ; 40
006105' 260 17 0 00 006120'             GO      MA31L1                  ; 100
006106' 260 17 0 00 006120'             GO      MA31L1                  ; 200
006107' 260 17 0 00 006130'             GO      MA31L2                  ; 1
006110' 260 17 0 00 006130'             GO      MA31L2                  ; 2
006111' 260 17 0 00 006130'             GO      MA31L2                  ; 4
006112' 260 17 0 00 006130'             GO      MA31L2                  ; 10
006113' 260 17 0 00 006130'             GO      MA31L2                  ; 20
006114' 260 17 0 00 006130'             GO      MA31L2                  ; 40
006115' 260 17 0 00 006130'             GO      MA31L2                  ; 100
006116' 260 17 0 00 006130'             GO      MA31L2                  ; 200
006117' 037 00 0 00 012161'             TMSG    <cleared after a CLRCCCODE command>

006120' 037 00 0 00 012615'     MA31L1: TMSG    <cleared after passing >
006121' 200 01 0 00 006073*             MOVE    1,TSTSUB
006122' 275 01 0 00 000002              SUBI    1,2
006123' 201 00 0 00 000001              MOVEI   1
006124' 242 00 0 01 000000              LSH     (1)
006125' 037 03 0 00 000000              PNT3
006126' 037 00 0 00 012622'             TMSGC   <  through PLI Input/Output buffers>
006127' 263 17 0 00 000000              RTN

006130' 037 00 0 00 012632'     MA31L2: TMSG    <set after passing >
006131' 200 01 0 00 006121*             MOVE    1,TSTSUB
006132' 275 01 0 00 000012              SUBI    1,^D10
006133' 201 00 0 00 000001              MOVEI   1
006134' 242 00 0 01 000000              LSH     (1)
006135' 037 03 0 00 000000              PNT3
006136' 037 00 0 00 012636'             TMSGC   <  through PLI Input/Output buffers with 'MPTESTPLIPAR' asserted>
006137' 263 17 0 00 000000              RTN
```

```

                                 ; Microcode:

                                 ;       CC is cleared initially

006140' 000000  010000           T31M:   MWORD   <ADDR=0,JMAP,J=1>                       ; 0
006141' 000000  000040
006142' 000152  000000                   MWORD   <CJP,J=5200,CENA,CCPP>                  ; 1
006143' 000400  150060
006144' 000251  000000                   MWORD   <JMAP,J=5100>                           ; 2
006145' 000000  000040

                                 ;       CC is cleared after passing floating 1's through
                                 ;         PLI Input/Output buffers

006146' 001005  000001                   MWORD   <ADDR=10,JMAP,J=500,SDO,OR,MGC=1,D=2>
006147' 732000  000040
006150' 001105  000002                   MWORD   <JMAP,J=500,SDO,OR,MGC=2,D=2>           ; 11
006151' 732000  000040
006152' 001205  000004                   MWORD   <JMAP,J=500,SDO,OR,MGC=4,D=2>           ; 12
006153' 732000  000040
006154' 001305  000010                   MWORD   <JMAP,J=500,SDO,OR,MGC=10,D=2>          ; 13
006155' 732000  000040
006156' 001405  000020                   MWORD   <JMAP,J=500,SDO,OR,MGC=20,D=2>          ; 14
006157' 732000  000040
006160' 001505  000040                   MWORD   <JMAP,J=500,SDO,OR,MGC=40,D=2>          ; 15
006161' 732000  000040
006162' 001605  000100                   MWORD   <JMAP,J=500,SDO,OR,MGC=100,D=2>         ; 16
006163' 732000  000040
006164' 001705  000200                   MWORD   <JMAP,J=500,SDO,OR,MGC=200,D=2>         ; 17
006165' 732000  000040

                                 ;       CC is set after passing floating 1's through
                                 ;         PLI Input/Output buffers with 'MPTESTPLIPAR'
                                 ;         asserted

006166' 003006  000001                   MWORD   <ADDR=30,JMAP,J=600,SDO,OR,MGC=1,D=2>   ; 30
006167' 732000  000040
006170' 003106  000002                   MWORD   <JMAP,J=600,SDO,OR,MGC=2,D=2>           ; 31
006171' 732000  000040
006172' 003206  000004                   MWORD   <JMAP,J=600,SDO,OR,MGC=4,D=2>           ; 32
006173' 732000  000040
006174' 003306  000010                   MWORD   <JMAP,J=600,SDO,OR,MGC=10,D=2>          ; 33
006175' 732000  000040
006176' 003406  000020                   MWORD   <JMAP,J=600,SDO,OR,MGC=20,D=2>          ; 34
006177' 732000  000040
006200' 003506  000040                   MWORD   <JMAP,J=600,SDO,OR,MGC=40,D=2>          ; 35
006201' 732000  000040
006202' 003606  000100                   MWORD   <JMAP,J=600,SDO,OR,MGC=100,D=2>         ; 36
006203' 732000  000040
006204' 003706  000200                   MWORD   <JMAP,J=600,SDO,OR,MGC=200,D=2>         ; 37
006205' 732000  000040

                                 ;       CC is cleared after a CLRCCCODE command

```

```
006206' 005000  510000          MWORD   <ADDR=50,JMAP,J=51>                             ; 50
006207' 000000  000040
006210' 005100  000020          MWORD   <CONT,SELM,MGC=20>                              ; 51
006211' 000000  002340
006212' 005252  000000          MWORD   <CJP,J=5200,CENA,CCPP>                          ; 52
006213' 000400  150060
006214' 005351  000000          MWORD   <JMAP,J=5100>                                   ; 53
006215' 000000  000040

                        ;       Routine to pass a data pattern through PLI buffers
                        ;       with good parity.

006216' 050005  012060          MWORD   <ADDR=500,JMAP,J=501,SOA,A=2,D=1,OENA,SELM,MGC=60>
006217' 401020  002040
006220' 050100  000006          MWORD   <CONT,SDO,D=1,SELP,MGC=6>                       ; 501
006221' 701000  001340
006222' 050200  000020          MWORD   <CONT,SDO,OR,B=2,D=2,SELF,MGC=20>               ; 502
006223' 732001  003340
006224' 050352  000000          MWORD   <CJP,J=5200,CENA,CCPP>                          ; 503
006225' 000400  150060
006226' 050451  000000          MWORD   <JMAP,J=5100>                                   ; 504
006227' 000000  000040

                        ;       Routine to pass a data pattern through PLI buffers
                        ;       with bad parity.

006230' 060006  012061          MWORD   <ADDR=600,JMAP,J=601,SOA,A=2,D=1,OENA,SELM,MGC=61>
006231' 401020  002040
006232' 060100  000006          MWORD   <CONT,SDO,D=1,SELP,MGC=6>                       ; 601
006233' 701000  001340
006234' 060200  000020          MWORD   <CONT,SDO,OR,B=2,D=2,SELF,MGC=20>               ; 602
006235' 732001  003340
006236' 060351  000000          MWORD   <CJP,J=5100,CENA,CCPP>                          ; 603
006237' 000400  150060
006240' 060452  000000          MWORD   <JMAP,J=5200>                                   ; 604
006241' 000000  000040

                        ; Subroutine - Passed - Write 1 to bit 12 of CSR register

006242' 510000  000000          MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>                ; 5100
006243' 742001  000340
006244' 510100  000000          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                     ; 5101
006245' 302001  000740
006246' 510200  260000          MWORD   <LDCT,J=26>                                     ; 5102
006247' 000000  000300
006250' 510351  030000          MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>              ; 5103
006251' 102021  000220
006252' 510400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>           ; 5104
006253' 431020  005340
006254' 510551  070040          MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>          ; 5105
006255' 001400  015060
006256' 510651  050000          MWORD   <JMAP,J=5105,D=1>                               ; 5106
006257' 001000  000040
006260' 510700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>              ; 5107
006261' 431010  005340
```

```
 006262' 511051  100000                    MWORD   <JMAP,J=5110,D=1>                        ; 5110
 006263' 001000  000040

                                   ; Subroutine - Failed - Write 1 to bit 13 of CSR register

 006264' 520000  000000                    MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>         ; 5200
 006265' 742001  000340
 006266' 520100  000000                    MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>              ; 5201
 006267' 302001  000740
 006270' 520200  250000                    MWORD   <LDCT,J=25>                              ; 5202
 006271' 000000  000300
 006272' 520352  030000                    MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>       ; 5203
 006273' 102021  000220
 006274' 520400  002004                    MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>    ; 5204
 006275' 431020  005340
 006276' 520552  070040                    MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>   ; 5205
 006277' 001400  015060
 006300' 520652  050000                    MWORD   <JMAP,J=5205,D=1>                        ; 5206
 006301' 001000  000040
 006302' 520700  000200                    MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>       ; 5207
 006303' 431010  005340
 006304' 521052  100000                    MWORD   <JMAP,J=5210,D=1>                        ; 5210
 006305' 001000  000040
 006306' 777777  777777                    -1
```

```
                                             ;#*********************************************************************
                                             ;* TEST 32 - CCChanErr Test
                                             ;
                                             ; Description:  Verify  that  condition  code 'CCChanErr'  can be
                                             ;               tested  by  the  microsequencer  for  conditional
                                             ;               branching, and that the condition code can be set
                                             ;               and reset by the hardware conditions which should
                                             ;               do so.
                                             ;
                                             ; Procedure:    Port Clear
                                             ;               Start up port
                                             ;               UC> perform an operation and then verify that
                                             ;                      CCChanErr is not asserted or asserted
                                             ;               UC> if passed - set bit 12 (DISABLE Complete)
                                             ;               UC> if failed - set bit 13 (ENABLE Complete)
                                             ;               Read CSR and determine test disposition
                                             ;
                                             ;               Repeat with several test cases:
                                             ;                 CC is cleared initially, after CBUS START is
                                             ;                    issued which does a RESET also
                                             ;                 CC is set after causing a 'short wc' error
                                             ;                 CC is set after causing a 'long wc' error
                                             ;                 CC is cleared after a successful data transfer
                                             ;
                                             ; Failure:      ---
                                             ;#*********************************************************************

                                             ; Test data

   006307' 254 00 0 00 006317'               TSTU32: JRST    TG32                    ; go start test
   006310' 200403  000032                            MPROC!NDMP!ZMPROC!32            ; test mask
   006311' 006420' 012654'                           T32M,,[ASCIZ ^CCChanErr Test^]
   006312' 012657' 012661'                           [EXP M6,MLAST!C6],,[EXP M7,C1,MLAST!C7]
   006313' 000000000000#                             TSTC3                           ; failure test table
   006314' 000000000000#                             TSTC4                           ;   ...
   006315' 000000  006603'                           TSTU33
   006316' 777777  777777                            -1

                                             ; Start test

   006317' 201 00 0 00 000000'               TG32:   MOVEI   Z8                      ; get address of module start
   006320' 260 17 0 00 005757*                       GO      TRACE                   ; handle trace output
   006321' 201 01 0 00 006420'                       MOVEI   1,T32M                  ; set up microcode address
   006322' 260 17 0 00 005761*                       GO      TLOAD                   ; load/verify it
   006323' 263 17 0 00 000000                        RTN                             ; failed - exit test
```

```

                                        ; Initialization

006324' 400 15 0 00 000000        TL32:   SETZ    ERFLG,                  ; clear error flag
006325' 201 06 0 00 006352'               MOVEI   6,TS32                  ; get test table address
006326' 402 00 0 00 006131*               SETZM   TSTSUB                  ; initialize subtest number
006327' 476 00 0 00 005766*               SETOM   TADDR                   ; clear start address
006330' 260 17 0 00 005767*               GO      IPACLR                  ; clear the CSR register
006331' 201 06 0 00 006352'               MOVEI   6,TS32                  ; get test table address
006332' 476 00 0 00 006327*               SETOM   TADDR                   ; clear start address
006333' 201 01 0 00 005000                MOVEI   1,5000                  ; get start address
006334' 242 01 0 00 000001                LSH     1,1                     ; position correctly
006335' 260 17 0 00 005244*               GO      LDRAR                   ; load the register
006336' 201 01 0 00 000010                MOVEI   1,MPRUN                 ; get 'MPRUN' bit
006337' 260 17 0 00 005246*               GO      LDCSR                   ; start up the port

                                        ; Loop on test execute table entries

006340' 260 17 0 00 006330*       TA32:   GO      IPACLR                  ; clear port
006341' 260 17 0 00 005770*       TB32:   GO      TEXEC                   ; execute table entry
006342' 254 00 0 00 006351'                 JRST  TX32                    ; end of sstep table
006343' 254 00 0 00 006341'                 JRST  TB32                    ; keep looping after call
006344' 474 15 0 00 000000                  SETO  ERFLG,                  ; error occurred

                                        ; Handle error printouts and scope looping

006345' 027 00 0 00 006406'               SCOPER  MA32                    ; print error message
006346' 254 00 0 00 006324'               JRST    TL32                    ; loop on error
006347' 254 00 0 00 006351'               JRST    TX32                    ; altmode exit
006350' 322 15 0 00 006340'               JUMPE   ERFLG,TA32              ; do next sstep table entry

                                        ; End of test

006351' 263 17 0 00 000000        TX32:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

                                        ; CC is cleared initially (after a MPSTARTCBUS)

006352' 140000  006373'           TS32:   TTABLE  (TCALL,TS32RD)          ; set up a write
006353' 040000  000000                    TTABLE  (TSTART,0)              ; start up port
006354' 240002  000000                    TTABLE  (TCHECK,2)              ; check if completed
006355' 340000  000000                    TTABLE  (TEXIT)                 ; exit if error yet

                                        ; CC is set after causing a 'short wc' error.

006356' 140000  006373'                   TTABLE  (TCALL,TS32RD)          ; set up a read
006357' 040000  000010                    TTABLE  (TSTART,10)             ; start up port
006360' 240002  000000                    TTABLE  (TCHECK,2)              ; check if completed
006361' 340000  000000                    TTABLE  (TEXIT)                 ; exit if error yet

```

```

                                     ; CC is set after causing a 'long wc' error.

006362' 140000  006373'                      TTABLE  (TCALL,TS32RD)          ; set up a read
006363' 040000  000030                       TTABLE  (TSTART,30)             ; start up port
006364' 240002  000000                       TTABLE  (TCHECK,2)              ; check if completed
006365' 340000  000000                       TTABLE  (TEXIT)                 ; exit if error yet

                                     ; CC is cleared after a successful data transfer

006366' 140000  006373'                      TTABLE  (TCALL,TS32RD)          ; set up a write
006367' 040000  000040                       TTABLE  (TSTART,40)             ; start up port
006370' 240002  000000                       TTABLE  (TCHECK,2)              ; check if completed
006371' 340000  000000                       TTABLE  (TEXIT)                 ; exit if error yet
006372' 000000  000000                       TTABLE  (TLAST)                 ; end of list

                                     ; Set up a read data transfer

006373' 201 14 0 00 000003           TS32RD: MOVEI   PAT,3                   ; get data pattern
006374' 201 01 0 00 000002                   MOVEI   1,2                     ; get transfer type
006375' 201 02 0 00 000002                   MOVEI   2,2                     ; get transfer length
006376' 260 17 0 00 002034*                  GO      BUFGEN                  ; generate buffer contents
006377' 200 01 0 00 011645'                  MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
006400' 260 17 0 00 005572*                  GO      CHINIT                  ; initialize software
006401' 551 01 0 00 005573*                  HRRZI   1,BUFF                  ; buffer address
006402' 201 02 0 00 000002                   MOVEI   2,2                     ; word count
006403' 400 03 0 00 000000                   SETZ    3,                      ; clear zero fill specifier
006404' 260 17 0 00 005576*                  GO      GENCCW                  ; generate a CCW list
006405' 263 17 0 00 000000                   RTN                             ; return

                                     ; Error messages

006406' 140000  012664'              MA32:   MSG!TXNOT![ASCIZ /Condition Code 'CC Chan Err' not /]
006407' 240000  006411'                      CALL!TXNOT!MA32LP               ; print subtest description
006410' 000000000000#                        LAST!CALL!TXALL!TTPNT           ; print test data

006411' 200 01 0 00 006326*          MA32LP: MOVE    1,TSTSUB                ; get subtest number
006412' 256 00 0 01 006413'                  XCT     MA32L-1(1)              ; print message
006413' 263 17 0 00 000000                   RTN                             ; return

006414' 037 00 0 00 012673'          MA32L:  TMSG    <cleared initially (after a 'MPSTARTCBUS')>
006415' 037 00 0 00 012704'                  TMSG    <set after causing a 'short wc' error>
006416' 037 00 0 00 012714'                  TMSG    <set after causing a 'long wc' error>
006417' 037 00 0 00 012724'                  TMSG    <cleared after a successful data transfer>
```

```

                                        ; Microcode:

                                        ;       CC is cleared initially, after CBUS START is
                                        ;       issued which does a RESET also.

006420' 000000  010000                  T32M:   MWORD   <ADDR=0,JMAP,J=1>               ; 0
006421' 000000  000040
006422' 000101  000220                          MWORD   <LDCT,J=100,SELC,MGC=220>       ; 1
006423' 000000  004300
006424' 000200  020000                          MWORD   <RPCT,J=2>                      ; 2
006425' 000000  000220
006426' 000352  000000                          MWORD   <CJP,J=5200,CENA,CCCE>          ; 3
006427' 000400  160060
006430' 000451  000000                          MWORD   <JMAP,J=5100>                   ; 4
006431' 000000  000040

                                        ;       CC is set after causing a 'short wc' error.

006432' 001000  110000                          MWORD   <ADDR=10,JMAP,J=11>             ; 10
006433' 000000  000040
006434' 001100  200220                          MWORD   <LDCT,J=20,SELC,MGC=220>        ; 11
006435' 000000  004300
006436' 001200  120000                          MWORD   <RPCT,J=12>                     ; 12
006437' 000000  000220
006440' 001310  000000                          MWORD   <CJS,J=1000,D=1>                ; 13
006441' 001000  000020
006442' 001410  000000                          MWORD   <CJS,J=1000,D=1>                ; 14
006443' 001000  000020
006444' 001510  000000                          MWORD   <CJS,J=1000,D=1>                ; 15
006445' 001000  000020
006446' 001601  000140                          MWORD   <LDCT,J=100,SELC,MGC=140>       ; 16
006447' 000000  004300
006450' 001700  170000                          MWORD   <RPCT,J=17>                     ; 17
006451' 000000  000220
006452' 002051  000000                          MWORD   <CJP,J=5100,CENA,CCCE>          ; 20
006453' 000400  160060
006454' 002152  000000                          MWORD   <JMAP,J=5200>                   ; 21
006455' 000000  000040

                                        ;       CC is set after causing a 'long wc' error.

006456' 003000  310000                          MWORD   <ADDR=30,JMAP,J=31>             ; 30
006457' 000000  000040
006460' 003100  200220                          MWORD   <LDCT,J=20,SELC,MGC=220>        ; 31
006461' 000000  004300
006462' 003200  320000                          MWORD   <RPCT,J=32>                     ; 32
006463' 000000  000220
006464' 003310  000000                          MWORD   <CJS,J=1000,D=1>                ; 33
006465' 001000  000020
006466' 003401  000140                          MWORD   <LDCT,J=100,SELC,MGC=140>       ; 34
006467' 000000  004300
006470' 003500  350000                          MWORD   <RPCT,J=35>                     ; 35
006471' 000000  000220
006472' 003651  000000                          MWORD   <CJP,J=5100,CENA,CCCE>          ; 36
```

```
006473' 000400  160060
006474' 003752  000000                 MWORD   <JMAP,J=5200>                           ; 37
006475' 000000  000040

                               ;       CC is cleared by a good transfer has occurred

006476' 004000  410000                 MWORD   <ADDR=40,JMAP,J=41>                     ; 40
006477' 000000  000040
006500' 004100  200220                 MWORD   <LDCT,J=20,SELC,MGC=220>                ; 41
006501' 000000  004300
006502' 004200  420000                 MWORD   <RPCT,J=42>                             ; 42
006503' 000000  000220
006504' 004310  000000                 MWORD   <CJS,J=1000,D=1>                        ; 43
006505' 001000  000020
006506' 004410  000000                 MWORD   <CJS,J=1000,D=1>                        ; 44
006507' 001000  000020
006510' 004500  000140                 MWORD   <CONT,SELC,MGC=140>                     ; 45
006511' 000000  004340
006512' 004652  000000                 MWORD   <CJP,J=5200,CENA,CCCE>                  ; 46
006513' 000400  160060
006514' 004751  000000                 MWORD   <JMAP,J=5100>                           ; 47
006515' 000000  000040

                               ;       Write a word over the CBUS

006516' 100010  020000                 MWORD   <ADDR=1000,CJP,J=1002,D=1,CENA,CCCA>    ; 1000
006517' 001400  000060
006520' 100110  000000                 MWORD   <JMAP,J=1000,D=1>                       ; 1001
006521' 001000  000040
006522' 100200  002040                 MWORD   <CONT,SOA,AND,D=2,OENA,SELF,MGC=40>     ; 1002
006523' 442000  003340
006524' 100300  060002                 MWORD   <LDCT,J=6,D=1,SELC,MGC=2>               ; 1003
006525' 001000  004300
006526' 100410  040000                 MWORD   <RPCT,J=1004>                           ; 1004
006527' 000000  000220
006530' 100500  000000                 MWORD   <CRTN>                                  ; 1005
006531' 000000  000240

                               ;       Initialization step

006532' 500050  010100                 MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>    ; 5000
006533' 000000  004040
006534' 500150  010030                 MWORD   <JMAP,J=5001,SELM,MGC=30>               ; 5001
006535' 000000  002040

                               ; Subroutine - Passed - Write 1 to bit 12 of CSR register

006536' 510000  000000                 MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>        ; 5100
006537' 742001  000340
006540' 510100  000000                 MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5101
006541' 302001  000740
006542' 510200  260000                 MWORD   <LDCT,J=26>                             ; 5102
006543' 000000  000300
006544' 510351  030000                 MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>      ; 5103
006545' 102021  000220
```

```
006546' 510400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>    ; 5104
006547' 431020  005340
006550' 510551  070040          MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>   ; 5105
006551' 001400  015060
006552' 510651  050000          MWORD   <JMAP,J=5105,D=1>                        ; 5106
006553' 001000  000040
006554' 510700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>       ; 5107
006555' 431010  005340
006556' 511051  100000          MWORD   <JMAP,J=5110,D=1>                        ; 5110
006557' 001000  000040

                        ; Subroutine - Failed - Write 1 to bit 13 of CSR register

006560' 520000  000000          MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>         ; 5200
006561' 742001  000340
006562' 520100  000000          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>              ; 5201
006563' 302001  000740
006564' 520200  250000          MWORD   <LDCT,J=25>                              ; 5202
006565' 000000  000300
006566' 520352  030000          MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>       ; 5203
006567' 102021  000220
006570' 520400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>    ; 5204
006571' 431020  005340
006572' 520552  070040          MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>   ; 5205
006573' 001400  015060
006574' 520652  050000          MWORD   <JMAP,J=5205,D=1>                        ; 5206
006575' 001000  000040
006576' 520700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>       ; 5207
006577' 431010  005340
006600' 521052  100000          MWORD   <JMAP,J=5210,D=1>                        ; 5210
006601' 001000  000040
006602' 777777  777777          -1
```

```
                                 ;#**********************************************************************
                                 ;* TEST 33 - CCCbLstWd Test
                                 ;
                                 ; Description:  Verify  that  condition  code 'CCCbLstWd'  can be
                                 ;               tested  by  the  microsequencer  for  conditional
                                 ;               branching, and that the condition code can be set
                                 ;               and reset by the hardware conditions which should
                                 ;               do so.
                                 ;
                                 ; Procedure:    Port Clear
                                 ;               Start up port
                                 ;               UC> perform an operation and then verify that
                                 ;                   CCCbLstWd is not asserted or asserted
                                 ;               UC> if passed - set bit 12 (DISABLE Complete)
                                 ;               UC> if failed - set bit 13 (ENABLE Complete)
                                 ;               Read CSR and determine test disposition
                                 ;
                                 ;               Repeat with several test cases:
                                 ;                 CC is initially cleared
                                 ;                 CC is set after final word of a 1 word transfer
                                 ;                 CC is not set after first word of a 2 word transfer
                                 ;                 CC is cleared after a CLRCCCODE command
                                 ;
                                 ; Failure:      ---
                                 ;#**********************************************************************

                                 ; Test data

006603' 254 00 0 00 006612'      TSTU33: JRST    TG33                     ; go start test
006604' 200403  000033                   MPROC!NDMP!ZMPROC!33             ; test mask
006605' 006702' 012735'                  T33M,,[ASCIZ ^CCCbLstWd Test^]
006606' 012657' 012661'                  [EXP M6,MLAST!C6],,[EXP M7,C1,MLAST!C7]
006607' 000000000000#                    TSTC3                            ; failure test table
006610' 000000000000#                    TSTC4                            ;   ...
006611' 777777  777777                   -1

                                 ; Start test

006612' 201 00 0 00 000000'      TG33:   MOVEI   Z8                       ; get address of module start
006613' 260 17 0 00 006320*              GO      TRACE                    ; handle trace output
006614' 201 01 0 00 006702'              MOVEI   1,T33M                   ; set up microcode address
006615' 260 17 0 00 006322*              GO      TLOAD                    ; load/verify it
006616' 263 17 0 00 000000               RTN                              ; failed - exit test

                                 ; Initialization

006617' 400 15 0 00 000000       TL33:   SETZ    ERFLG,                   ; clear error flag
006620' 201 06 0 00 006635'              MOVEI   6,TS33                   ; get test table address
006621' 402 00 0 00 006411*              SETZM   TSTSUB                   ; initialize subtest number
006622' 476 00 0 00 006332*              SETOM   TADDR                    ; clear start address
```

```

                                  ; Loop on test execute table entries

 006623' 260 17 0 00 006340*      TA33:   GO      IPACLR                  ; clear port
 006624' 260 17 0 00 006341*      TB33:   GO      TEXEC                   ; execute table entry
 006625' 254 00 0 00 006634'               JRST   TX33                    ; end of sstep table
 006626' 254 00 0 00 006624'               JRST   TB33                    ; keep looping after call
 006627' 474 15 0 00 000000                SETO   ERFLG,                  ; error occurred

                                  ; Handle error printouts and scope looping

 006630' 027 00 0 00 006670'              SCOPER  MA33                    ; print error message
 006631' 254 00 0 00 006617'              JRST    TL33                    ; loop on error
 006632' 254 00 0 00 006634'              JRST    TX33                    ; altmode exit
 006633' 322 15 0 00 006623'              JUMPE   ERFLG,TA33              ; do next sstep table entry

                                  ; End of test

 006634' 263 17 0 00 000000       TX33:   RTN                             ; return

                                  ; Test Execute Table, as:  (CMD,parameters)

                                  ; CC is cleared initially

 006635' 140000  005305'          TS33:   TTABLE  (TCALL,T27SR1)          ; set up read of 1 word
 006636' 040000  000000                   TTABLE  (TSTART,0)              ; start up port
 006637' 240002  000000                   TTABLE  (TCHECK,2)              ; check if completed
 006640' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

                                  ; CC is set after final word of a 1 word transfer

 006641' 140000  005305'                  TTABLE  (TCALL,T27SR1)          ; set up read of 1 word
 006642' 040000  000000                   TTABLE  (TSTART,0)              ; start up port
 006643' 240002  000000                   TTABLE  (TCHECK,2)              ; check if completed
 006644' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

                                  ; CC is not set after first word of a 2 word transfer

 006645' 140000  005307'                  TTABLE  (TCALL,T27SR2)          ; set up read of 2 words
 006646' 040000  000000                   TTABLE  (TSTART,0)              ; start up port
 006647' 240002  000000                   TTABLE  (TCHECK,2)              ; check if completed
 006650' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

                                  ; CC is cleared after a CLRCCCODE command

 006651' 140000  005305'                  TTABLE  (TCALL,T27SR1)          ; set up read of 1 word
 006652' 040000  000000                   TTABLE  (TSTART,0)              ; start up port
 006653' 240002  000000                   TTABLE  (TCHECK,2)              ; check if completed
 006654' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet
 006655' 000000  000000                   TTABLE  (TLAST)                 ; end of list

```

```
                                        ; Routine to set up data transfer

006656' 201 02 0 00 000001             T33SR1: MOVEI   2,1                     ; 1 word transfer
006657' 334 00 0 00 000000                     SKIPA                           ; continue
006660' 201 02 0 00 000002             T33SR2: MOVEI   2,2                     ; 2 word transfer
006661' 402 00 0 00 006401*                    SETZM   BUFF                    ; set data to 0's
006662' 200 01 0 00 011645'                    MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
006663' 260 17 0 00 006400*                    GO      CHINIT                  ; initialize software
006664' 551 01 0 00 006661*                    HRRZI   1,BUFF                  ; buffer address
006665' 400 03 0 00 000000                     SETZ    3,                      ; clear zero fill specifier
006666' 260 17 0 00 006404*                    GO      GENCCW                  ; generate a CCW list
006667' 263 17 0 00 000000                     RTN                             ; return

                                        ; Error messages

006670' 140000  012740'                MA33:   MSG!TXNOT![ASCIZ /Condition Code 'CC Cb Lst Wd' not /]
006671' 240000  006673'                        CALL!TXNOT!MA33LP               ; print subtest description
006672' 000000000000#                          LAST!CALL!TXALL!TTPNT           ; print test data

006673' 200 01 0 00 006621*            MA33LP: MOVE    1,TSTSUB                ; get subtest number
006674' 256 00 0 01 006675'                    XCT     MA33L-1(1)              ; print message
006675' 263 17 0 00 000000                     RTN                             ; return

006676' 037 00 0 00 012425'            MA33L:  TMSG    <initially cleared>
006677' 037 00 0 00 012747'                    TMSG    <set after final word of a 1 word transfer>
006700' 037 00 0 00 012760'                    TMSG    <cleared after first word of a 2 word transfer>
006701' 037 00 0 00 012161'                    TMSG    <cleared after a CLRCCCODE command>

                                        ; Microcode:

                                        ;       CC is cleared initially

006702' 000000  010000                 T33M:   MWORD   <ADDR=0,JMAP,J=1>                       ; 0
006703' 000000  000040
006704' 000100  000200                         MWORD   <CONT,SELC,MGC=200>                     ; 1
006705' 000000  004340
006706' 000252  000000                         MWORD   <CJP,J=5200,CENA,CCLW>                  ; 2
006707' 000400  170060
006710' 000351  000000                         MWORD   <JMAP,J=5100>                           ; 3
006711' 000000  000040

                                        ;       CC is set after final word of a 1 word transfer

006712' 001000  110000                         MWORD   <ADDR=10,JMAP,J=11,SDO,AND,D=2>         ; 10
006713' 742000  000040
006714' 001100  000200                         MWORD   <CONT,D=1,SELC,MGC=200>                 ; 11 (start CBUS)
006715' 001000  004340
006716' 001200  140000                         MWORD   <CJP,J=14,D=1,CENA,CCCA>                ; 12 (wait till avail)
006717' 001400  000060
006720' 001300  120000                         MWORD   <JMAP,J=12,D=1>                         ; 13
006721' 001000  000040
006722' 001451  000000                         MWORD   <CJP,J=5100,CENA,CCLW>                  ; 14
006723' 000400  170060
006724' 001552  000000                         MWORD   <JMAP,J=5200>                           ; 15
```

```
006725' 000000  000040

                                ;       CC is not set after first word of a 2 word transfer

006726' 003000  310000                  MWORD   <ADDR=30,JMAP,J=31,SDO,AND,D=2>              ; 30
006727' 742000  000040
006730' 003100  000200                  MWORD   <CONT,D=1,SELC,MGC=200>                      ; 31 (start CBUS)
006731' 001000  004340
006732' 003200  340000                  MWORD   <CJP,J=34,D=1,CENA,CCCA>                     ; 32 (wait till avail)
006733' 001400  000060
006734' 003300  320000                  MWORD   <JMAP,J=32,D=1>                              ; 33
006735' 001000  000040
006736' 003452  000000                  MWORD   <CJP,J=5200,CENA,CCLW>                       ; 34
006737' 000400  170060
006740' 003551  000000                  MWORD   <JMAP,J=5100>                                ; 35
006741' 000000  000040

                                ;       CC is cleared after a CLRCCCODE command

006742' 005000  510000                  MWORD   <ADDR=50,JMAP,J=51>                          ; 50
006743' 000000  000040
006744' 005100  000200                  MWORD   <CONT,D=1,SELC,MGC=200>                      ; 51 (start CBUS)
006745' 001000  004340
006746' 005200  540000                  MWORD   <CJP,J=54,D=1,CENA,CCCA>                     ; 52 (wait till avail)
006747' 001400  000060
006750' 005300  120000                  MWORD   <JMAP,J=12,D=1>                              ; 53
006751' 001000  000040
006752' 005400  000020                  MWORD   <CONT,SELM,MGC=20>                           ; 54
006753' 000000  002340
006754' 005552  000000                  MWORD   <CJP,J=5200,CENA,CCLW>                       ; 55
006755' 000400  170060
006756' 005651  000000                  MWORD   <JMAP,J=5100>                                ; 56
006757' 000000  000040

                                ; Subroutine - Passed - Write 1 to bit 12 of CSR register

006760' 510000  000000                  MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>             ; 5100
006761' 742001  000340
006762' 510100  000000                  MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                  ; 5101
006763' 302001  000740
006764' 510200  260000                  MWORD   <LDCT,J=26>                                  ; 5102
006765' 000000  000300
006766' 510351  030000                  MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>           ; 5103
006767' 102021  000220
006770' 510400  002004                  MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>        ; 5104
006771' 431020  005340
006772' 510551  070040                  MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>       ; 5105
006773' 001400  015060
006774' 510651  050000                  MWORD   <JMAP,J=5105,D=1>                            ; 5106
006775' 001000  000040
006776' 510700  000200                  MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>           ; 5107
006777' 431010  005340
007000' 511051  100000                  MWORD   <JMAP,J=5110,D=1>                            ; 5110
007001' 001000  000040
```

```
                                             ; Subroutine - Failed - Write 1 to bit 13 of CSR register

007002' 520000  000000                              MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>        ; 5200
007003' 742001  000340
007004' 520100  000000                              MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5201
007005' 302001  000740
007006' 520200  250000                              MWORD   <LDCT,J=25>                             ; 5202
007007' 000000  000300
007010' 520352  030000                              MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>      ; 5203
007011' 102021  000220
007012' 520400  002004                              MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5204
007013' 431020  005340
007014' 520552  070040                              MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>  ; 5205
007015' 001400  015060
007016' 520652  050000                              MWORD   <JMAP,J=5205,D=1>                       ; 5206
007017' 001000  000040
007020' 520700  000200                              MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5207
007021' 431010  005340
007022' 521052  100000                              MWORD   <JMAP,J=5210,D=1>                       ; 5210
007023' 001000  000040
007024' 777777  777777                              -1
```

```
                                        ;#**********************************************************************
                                        ;* TEST 34 - Basic Local Store Test
                                        ;
                                        ; Description:  This test writes data to local storage and  reads
                                        ;               it back and verifies it.  It is a basic  test  in
                                        ;               that only data patterns 0's and 1's are used  and
                                        ;               only a few microinstructions are executed to read
                                        ;               and write a location.
                                        ;
                                        ; Procedure:    Port Clear
                                        ;               Load microwords with local storage addresses
                                        ;               Set start address
                                        ;               Single step 3 times
                                        ;               Read EBUF and verify data 0's
                                        ;               Set start address
                                        ;               Single step 3 times
                                        ;               Read EBUF and verify data 1's
                                        ;
                                        ;               Repeat above steps for addresses 0-1024
                                        ;
                                        ; Failure:      ---
                                        ;#**********************************************************************

                                        ; Test data

007025' 254 00 0 00 007035'             TSTU34: JRST    TG34                    ; go start test
007026' 210403  000034                          MPROC!MBUS!NDMP!ZMPROC!34       ; test mask
007027' 007122' 012772'                         T34M,,[ASCIZ ^Basic Local Store Test^]
007030' 012777' 013001'                         [EXP M13,MLAST!M2],,[EXP M14,M15,MLAST!M11]
007031' 000000  007137'                         TSTU35                  ; failure test table
007032' 000000  010326'                         TSTU36                  ;   ...
007033' 000000  010521'                         TSTU37
007034' 777777  777777                          -1

                                        ; Start test

007035' 201 00 0 00 000000'             TG34:   MOVEI   Z8                      ; get address of module start
007036' 260 17 0 00 006613*                     GO      TRACE                   ; handle trace output
007037' 201 01 0 00 007122'                     MOVEI   1,T34M                  ; set up microcode address
007040' 260 17 0 00 006615*                     GO      TLOAD                   ; load/verify it
007041' 263 17 0 00 000000                      RTN                             ; failed - exit test

                                        ; Initialization

007042' 400 15 0 00 000000              TL34:   SETZ    ERFLG,                  ; clear error flag
007043' 260 17 0 00 006623*                     GO      IPACLR                  ; clear port
007044' 402 00 0 00 006673*                     SETZM   TSTSUB                  ; initialize subtest number
007045' 201 06 0 00 007057'                     MOVEI   6,TS34                  ; get sstep table address
```

```

                                        ; Loop on single step table entries

007046' 260 17 0 00 000000*             TA34:   GO      BEXEC                   ; execute table entry
007047' 254 00 0 00 007056'                     JRST    TX34                    ; end of sstep table
007050' 254 00 0 00 007046'                     JRST    TA34                    ; keep looping after call
007051' 474 15 0 00 000000                      SETO    ERFLG,                  ; error - EBUF data incorrect

                                        ; Handle error printouts and scope looping

007052' 027 00 0 00 007113'                     SCOPER  MA34                    ; print error message
007053' 254 00 0 00 007042'                     JRST    TL34                    ; loop on error
007054' 254 00 0 00 007056'                     JRST    TX34                    ; altmode exit
007055' 322 15 0 00 007046'                     JUMPE   ERFLG,TA34              ; do next sstep table entry

                                        ; End of test

007056' 263 17 0 00 000000              TX34:   RTN                             ; return

                                        ; Sstep table, with EBUF data, as:  (CMD,#Steps,Start addr,End Addr,EBUF data)

007057' 300000  007067'                 TS34:   ATABLE  (SSCALL,TS34IN)         ; initialize test parameters
007060' 400000  007071'                 TS34L:  ATABLE  (SSCHK,TS34ST)          ; set up single step parameters
007061' 100300  000003                          ATABLE  (SSSTRT,3,0,3,0)        ; do 3 single steps/check results
007062' 000000  000000
007063' 100300  030006                          ATABLE  (SSSTRT,3,3,6,-1)       ; do 3 single steps/check results
007064' 777777  777777
007065' 500000  007060'                         ATABLE  (SSJRST,TS34L)          ; do next address
007066' 000000  000000                          ATABLE  (SSLAST)                ; end of table

                                        ; Special routines

007067' 476 00 0 00 011165'             TS34IN: SETOM   SAVDAT                  ; initialize address
007070' 263 17 0 00 000000                      RTN                             ; return

007071' 350 07 0 00 011165'             TS34ST: AOS     7,SAVDAT                ; point to next address
007072' 305 07 0 00 002000                      CAIGE   7,^D1024                ; done yet?
007073' 350 00 0 17 000000                      AOS     (P)                     ; yes - set up RTN+2
007074' 332 00 0 00 000172*                     SKIPE   UDEBUG                  ; debug mode?
                                                JRST    [CAIGE  7,10            ; yes - limit to 8 times
                                                         AOS    (P)
007075' 254 00 0 00 013004'                              JRST   .+1]
007076' 135 07 0 00 013007'                     LDB     7,[POINT 10,SAVDAT,35]  ; get bits 00-09
007077' 137 07 0 00 013010'                     DPB     7,[POINT 10,T34M,35]    ; save in 1st microword
007100' 137 07 0 00 013011'                     DPB     7,[POINT 10,T34M+2,35]  ; save in 2nd microword
007101' 137 07 0 00 013012'                     DPB     7,[POINT 10,T34MA,35]   ; save in 3rd microword
007102' 137 07 0 00 013013'                     DPB     7,[POINT 10,T34MA+2,35] ; save in 4th microword
007103' 316 16 0 00 000000*                     CAMN    MBCN,PORTNI             ; NI port?
007104' 476 00 0 00 000000*                     SETOM   TSLOD1                  ; yes - must reload microcode each time
007105' 316 16 0 00 000000*                     CAMN    MBCN,PORTCI             ; CI port?
007106' 476 00 0 00 000000*                     SETOM   TSLOD2                  ; yes - must reload microcode each time
007107' 201 01 0 00 007122'                     MOVEI   1,T34M                  ; get microcode address
007110' 260 17 0 00 007040*                     GO      TLOAD                   ; load it
007111' 370 00 0 17 000000                      SOS     (P)                     ; error - exit test
007112' 263 17 0 00 000000                      RTN                             ; return
```

```

                                        ; Error messages

007113' 240000  007115'                 MA34:   CALL!TXNOT!MA34PN               ; print data
007114' 000000000000#                           LAST!CALL!TXALL!BBPNT           ; print sequencer data

007115' 037 00 0 00 013014'             MA34PN: TMSGC   <Wrote data pattern to local storage addr >
007116' 200 00 0 00 011165'                     MOVE    SAVDAT#                 ; get address
007117' 037 16 0 00 000003                      PNTOCS                          ; print it
007120' 037 00 0 00 013025'                     TMSGC   <Then read back and placed in EBUF>
007121' 263 17 0 00 000000                      RTN

                                        ; Microcode:

007122' 000000  012000                  T34M:   MWORD   <ADDR=0,JMAP,J=1,SOA,AND,D=1,OENA,LDLM,MGC=0>   ; 0
007123' 441000  230040
007124' 000100  000000                          MWORD   <CONT,SDO,OR,D=2,RDLM,MGC=0>                    ; 1
007125' 732000  220340
007126' 000200  002004                          MWORD   <CONT,SOA,OR,D=1,OENA,SELE,MGC=4>               ; 2
007127' 431000  005340

007130' 000300  042000                  T34MA:  MWORD   <JMAP,J=4,SAB,XNOR,D=1,OENA,LDLM,MGC=0>         ; 3
007131' 171000  230040
007132' 000400  000000                          MWORD   <CONT,SDO,OR,D=2,RDLM,MGC=0>                    ; 4
007133' 732000  220340
007134' 000500  002004                          MWORD   <CONT,SOA,OR,D=1,OENA,SELE,MGC=4>               ; 5
007135' 431000  005340
007136' 777777  777777                          -1
```

```
                                      ;#**********************************************************************
                                      ;* TEST 35 - Local Store Data Test
                                      ;
                                      ; Description:  This test writes data to local storage and  reads
                                      ;               it back and verifies it. It builds a data pattern
                                      ;               then  writes  and  reads  all  1024 local storage
                                      ;               locations.  This is repeated until  all  72  data
                                      ;               patterns have been done.
                                      ;
                                      ; Procedure:    Port Clear
                                      ;               Load microcode
                                      ;               Start up port
                                      ;               UC> build a data pattern
                                      ;               UC> write/read all of local storage with this
                                      ;                   data pattern
                                      ;               UC> do data compare
                                      ;               If failed - read R10 to get correct data
                                      ;                           read R11 to get actual data
                                      ;                           read R12 to get part of LS addr
                                      ;                           read R13 to get rest of LS addr
                                      ;                           halt at location 100
                                      ;               If passed - do next data pattern until done,
                                      ;                 then halt at 157
                                      ;
                                      ; Failure:      ---
                                      ;#**********************************************************************

                                      ; Test data

007137' 254 00 0 00 007146'           TSTU35: JRST    TG35                    ; go start test
007140' 200403  000035                        MPROC!NDMP!ZMPROC!35            ; test mask
007141' 007243' 013035'                       T35M,,[ASCIZ ^Local Store Data Test^]
007142' 012777' 013001'                       [EXP M13,MLAST!M2],,[EXP M14,M15,MLAST!M11]
007143' 000000  010326'                       TSTU36                          ; failure test table
007144' 000000  010521'                       TSTU37                          ;   ...
007145' 777777  777777                        -1

                                      ; Start test

007146' 201 00 0 00 000000'           TG35:   MOVEI   Z8                      ; get address of module start
007147' 260 17 0 00 007036*                   GO      TRACE                   ; handle trace output
007150' 201 01 0 00 007243'                   MOVEI   1,T35M                  ; set up microcode address
007151' 260 17 0 00 007110*                   GO      TLOAD                   ; load/verify it
007152' 263 17 0 00 000000                    RTN                             ; failed - exit test
```

```

                                        ; Initialization

007153' 400 15 0 00 000000      TL35:   SETZ    ERFLG,                  ; clear error flag
007154' 260 17 0 00 007043*             GO      IPACLR                  ; clear port
007155' 402 00 0 00 007044*             SETZM   TSTSUB                  ; initialize subtest number
007156' 201 01 0 00 000200              MOVEI   1,200                   ; allow 200 msecs
007157' 201 02 0 00 000100              MOVEI   2,100                   ; correct error address
007160' 201 03 0 00 000157              MOVEI   3,157                   ; correct final address
007161' 201 04 0 00 005777              MOVEI   4,5777                  ; start address
007162' 260 17 0 00 001666*             GO      AEXEC                   ; do the test
007163' 474 15 0 00 000000              SETO    ERFLG,                  ; error - set error flag

                                        ; Handle error printouts and scope looping

007164' 027 00 0 00 007170'                     SCOPER  MA35            ; print error message
007165' 254 00 0 00 007153'                     JRST    TL35            ; loop on error
007166' 254 00 0 00 007167'                     JRST    TX35            ; altmode exit

                                        ; End of test

007167' 263 17 0 00 000000      TX35:   RTN                             ; return

                                        ; Error messages

007170' 140000  013042'         MA35:   MSG!TXNOT![ASCIZ /Local storage test error:/]
007171' 000000000000#                   CALL!TXALL!AAPNT                ; print sequencer data
007172' 270000  007173'                 LAST!CALL!TXALL!MA35PN          ; get/print error data

007173' 201 02 0 00 000200      MA35PN: MOVEI   2,200                   ; initial start address
007174' 200 00 0 00 000002      MA35P1: MOVE    2                       ; get start address
007175' 260 17 0 00 006335*             GO      LDRAR                   ; write RAR
007176' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get 'MPRUN'
007177' 260 17 0 00 006337*             GO      LDCSR                   ; write CSR
007200' 201 01 0 00 200000              MOVEI   1,TSTEBF                ; get 'TSTEBF'
007201' 260 17 0 00 007177*             GO      LDCSR                   ; write CSR
007202' 260 17 0 00 000000*             GO      RDEBUF                  ; read EBUF
007203' 202 00 0 02 007037'             MOVEM   SAVDA1-200(2)           ; save data
007204' 350 00 0 00 000002              AOS     2                       ; increment address
007205' 307 02 0 00 000103              CAIG    2,103                   ; done yet?
007206' 254 00 0 00 007174'             JRST    MA35P1                  ; no - continue

007207' 200 01 0 00 001114*             MOVE    1,SCOSW                 ; get switches
007210' 603 01 0 00 000200              TLNE    1,TXTINH                ; full printout?
007211' 037 00 0 00 013050'             TMSGC   <LS>                    ; no
007212' 607 01 0 00 000200              TLNN    1,TXTINH                ; full printout?
007213' 037 00 0 00 013051'             TMSGC   <LS address:  >         ; yes
007214' 200 00 0 00 007241'             MOVE    SAVDA1+2
007215' 200 01 0 00 007242'             MOVE    1,SAVDA1+3
007216' 242 01 0 00 000040              LSH     1,^D32
007217' 246 01 0 00 000004              LSHC    1,4
007220' 037 04 0 00 000000              PNT4
```

```
007221' 200 01 0 00 007207*                     MOVE    1,SCOSW                 ; get switches
007222' 603 01 0 00 000200                      TLNE    1,TXTINH                ; full printout?
007223' 037 00 0 00 013055'                     TMSG    <(C):  >                ; no
007224' 607 01 0 00 000200                      TLNN    1,TXTINH                ; full printout?
007225' 037 00 0 00 013057'                     TMSGC   <Data:  Correct >
007226' 200 00 0 00 007237'                     MOVE    SAVDA1
007227' 037 13 0 00 000000                      PNTHW
007230' 603 01 0 00 000200                      TLNE    1,TXTINH                ; full printout?
007231' 037 00 0 00 013063'                     TMSGC   <        (A):  >        ; no
007232' 607 01 0 00 000200                      TLNN    1,TXTINH                ; full printout?
007233' 037 00 0 00 013067'                     TMSGC   <          Actual >     ; yes
007234' 200 00 0 00 007240'                     MOVE    SAVDA1+1
007235' 037 13 0 00 000000                      PNTHW
007236' 263 17 0 00 000000                      RTN

007237'                                 SAVDA1: BLOCK   4                       ; data area

                                        ; Microcode:

007243' 577700  000000                  T35M:   MWORD   <ADDR=5777,JZ>                          ; 5777
007244' 000000  000000

                                        ;       Build data patterns

007245' 000000  000000                          MWORD   <ADDR=0,CONT,SDO,AND,B=7,D=2>           ; 0
007246' 742003  400340
007247' 000110  000000                          MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>        ; 1 - 0's
007250' 432074  000020
007251' 000210  000000                          MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>      ; 2 - comp
007252' 472074  000020
007253' 001300  000000                          MWORD   <CONT,SOA,PLUS,CRY,A=7,B=7,D=2>         ; 3
007254' 402073  400740
007255' 000410  000000                          MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>        ; 4 - 1
007256' 432074  000020
007257' 000510  000000                          MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>      ; 5 - comp
007260' 472074  000020
007261' 000600  000000                          MWORD   <CONT,SOA,OR,A=7,B=7,D=7>               ; 6
007262' 437073  400340
007263' 000710  000000                          MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>        ; 7 - 2
007264' 432074  000020
007265' 001010  000000                          MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>      ; 10 - comp
007266' 472074  000020
007267' 001100  000000                          MWORD   <CONT,SOA,OR,A=7,B=7,D=7>               ; 11
007270' 437073  400340
007271' 001210  000000                          MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>        ; 12 - 4
007272' 432074  000020
007273' 001310  000000                          MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>      ; 13 - comp
007274' 472074  000020
007275' 001400  000000                          MWORD   <CONT,SOA,OR,A=7,B=7,D=7>               ; 14
007276' 437073  400340
007277' 001510  000000                          MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>        ; 15 - 10
007300' 432074  000020
007301' 001610  000000                          MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>      ; 16 - comp
007302' 472074  000020
007303' 001700  000000                          MWORD   <CONT,SOA,OR,A=7,B=7,D=7>               ; 17
```

```
 5977  007304' 437073  400340
 5978  007305' 002010  000000             MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 20 - 20
 5979  007306' 432074  000020
 5980  007307' 002110  000000             MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 21 - comp
 5981  007310' 472074  000020
 5982  007311' 002200  000000             MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 22
 5983  007312' 437073  400340
 5984  007313' 002310  000000             MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 23 - 40
 5985  007314' 432074  000020
 5986  007315' 002410  000000             MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 24 - comp
 5987  007316' 472074  000020
 5988  007317' 002500  000000             MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 25
 5989  007320' 437073  400340
 5990  007321' 002610  000000             MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 26 - 100
 5991  007322' 432074  000020
 5992  007323' 002710  000000             MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 27 - comp
 5993  007324' 472074  000020
 5994  007325' 003000  000000             MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 30
 5995  007326' 437073  400340
 5996  007327' 003110  000000             MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 31 - 200
 5997  007330' 432074  000020
 5998  007331' 003210  000000             MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 32 - comp
 5999  007332' 472074  000020
 6000  007333' 003300  000000             MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 33
 6001  007334' 437073  400340
 6002  007335' 003410  000000             MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 34 - 400
 6003  007336' 432074  000020
 6004  007337' 003510  000000             MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 35 - comp
 6005  007340' 472074  000020
 6006  007341' 003600  000000             MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 36
 6007  007342' 437073  400340
 6008  007343' 003710  000000             MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 37 - 1000
 6009  007344' 432074  000020
 6010  007345' 004010  000000             MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 40 - comp
 6011  007346' 472074  000020
 6012  007347' 004100  000000             MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 41
 6013  007350' 437073  400340
 6014  007351' 004210  000000             MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 42 - 2000
 6015  007352' 432074  000020
 6016  007353' 004310  000000             MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 43 - comp
 6017  007354' 472074  000020
 6018  007355' 004400  000000             MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 44
 6019  007356' 437073  400340
 6020  007357' 004510  000000             MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 45 - 4000
 6021  007360' 432074  000020
 6022  007361' 004610  000000             MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 46 - comp
 6023  007362' 472074  000020
 6024  007363' 004700  000000             MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 47
 6025  007364' 437073  400340
 6026  007365' 005010  000000             MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 50 - 10000
 6027  007366' 432074  000020
 6028  007367' 005110  000000             MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 51 - comp
 6029  007370' 472074  000020
 6030  007371' 005200  000000             MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 52
 6031  007372' 437073  400340
```

```
6032  007373' 005310  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 53 - 20000
6033  007374' 432074  000020
6034  007375' 005410  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 54 - comp
6035  007376' 472074  000020
6036  007377' 005500  000000          MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 55
6037  007400' 437073  400340
6038  007401' 005610  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 56 - 40000
6039  007402' 432074  000020
6040  007403' 005710  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 57 - comp
6041  007404' 472074  000020
6042  007405' 006000  000000          MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 60
6043  007406' 437073  400340
6044  007407' 006110  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 61 - 100000
6045  007410' 432074  000020
6046  007411' 006210  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 62 - comp
6047  007412' 472074  000020
6048  007413' 006300  000000          MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 63
6049  007414' 437073  400340
6050  007415' 006410  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 64 - 200000
6051  007416' 432074  000020
6052  007417' 006510  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 65 - comp
6053  007420' 472074  000020
6054  007421' 006600  000000          MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 66
6055  007422' 437073  400340
6056  007423' 006710  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 67 - 400000
6057  007424' 432074  000020
6058  007425' 007010  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 70 - comp
6059  007426' 472074  000020
6060  007427' 007100  000000          MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 71
6061  007430' 437073  400340
6062  007431' 007210  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 72 - 1,,0
6063  007432' 432074  000020
6064  007433' 007310  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 73 - comp
6065  007434' 472074  000020
6066  007435' 007400  000000          MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 74
6067  007436' 437073  400340
6068  007437' 007510  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 75 - 2,,0
6069  007440' 432074  000020
6070  007441' 007610  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 76 - comp
6071  007442' 472074  000020
6072  007443' 007700  000000          MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 77
6073  007444' 437073  400340
6074  007445' 010010  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 100 - 4,,0
6075  007446' 432074  000020
6076  007447' 010110  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 101 - comp
6077  007450' 472074  000020
6078  007451' 010200  000000          MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 102
6079  007452' 437073  400340
6080  007453' 010310  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 103 - 10,,0
6081  007454' 432074  000020
6082  007455' 010410  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 104 - comp
6083  007456' 472074  000020
6084  007457' 010500  000000          MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 105
6085  007460' 437073  400340
6086  007461' 010610  000000          MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 106 - 20,,0
```

```
 6087  007462' 432074  000020
 6088  007463' 010710  000000                    MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 107 - comp
 6089  007464' 472074  000020
 6090  007465' 011000  000000                    MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 110
 6091  007466' 437073  400340
 6092  007467' 011110  000000                    MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 111 - 40,,0
 6093  007470' 432074  000020
 6094  007471' 011210  000000                    MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 112 - comp
 6095  007472' 472074  000020
 6096  007473' 011300  000000                    MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 113
 6097  007474' 437073  400340
 6098  007475' 011410  000000                    MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 114 - 100,,0
 6099  007476' 432074  000020
 6100  007477' 011510  000000                    MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 115 - comp
 6101  007500' 472074  000020
 6102  007501' 011600  000000                    MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 116
 6103  007502' 437073  400340
 6104  007503' 011710  000000                    MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 117 - 200,,0
 6105  007504' 432074  000020
 6106  007505' 012010  000000                    MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 120 - comp
 6107  007506' 472074  000020
 6108  007507' 012100  000000                    MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 121
 6109  007510' 437073  400340
 6110  007511' 012210  000000                    MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 122 - 400,,0
 6111  007512' 432074  000020
 6112  007513' 012310  000000                    MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 123 - comp
 6113  007514' 472074  000020
 6114  007515' 012400  000000                    MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 124
 6115  007516' 437073  400340
 6116  007517' 012510  000000                    MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 125 - 1000,,0
 6117  007520' 432074  000020
 6118  007521' 012610  000000                    MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 126 - comp
 6119  007522' 472074  000020
 6120  007523' 012700  000000                    MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 127
 6121  007524' 437073  400340
 6122  007525' 013010  000000                    MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 130 - 2000,,0
 6123  007526' 432074  000020
 6124  007527' 013110  000000                    MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 131 - comp
 6125  007530' 472074  000020
 6126  007531' 013200  000000                    MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 132
 6127  007532' 437073  400340
 6128  007533' 013310  000000                    MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 133 - 4000,,0
 6129  007534' 432074  000020
 6130  007535' 013410  000000                    MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 134 - comp
 6131  007536' 472074  000020
 6132  007537' 013500  000000                    MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 135
 6133  007540' 437073  400340
 6134  007541' 013610  000000                    MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 136 - 10000,,0
 6135  007542' 432074  000020
 6136  007543' 013710  000000                    MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>        ; 137 - comp
 6137  007544' 472074  000020
 6138  007545' 014000  000000                    MWORD   <CONT,SOA,OR,A=7,B=7,D=7>                 ; 140
 6139  007546' 437073  400340
 6140  007547' 014110  000000                    MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>          ; 141 - 20000,,0
 6141  007550' 432074  000020
 6142  007551' 014210  000000                  MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>      ; 142 - comp
 6143  007552' 472074  000020
 6144  007553' 014300  000000                  MWORD   <CONT,SOA,OR,A=7,B=7,D=7>               ; 143
 6145  007554' 437073  400340
 6146  007555' 014410  000000                  MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>        ; 144 - 40000,,0
 6147  007556' 432074  000020
 6148  007557' 014510  000000                  MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>      ; 145 - comp
 6149  007560' 472074  000020
 6150  007561' 014600  000000                  MWORD   <CONT,SOA,OR,A=7,B=7,D=7>               ; 146
 6151  007562' 437073  400340
 6152  007563' 014710  000000                  MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>        ; 147 - 100000,,0
 6153  007564' 432074  000020
 6154  007565' 015010  000000                  MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>      ; 150 - comp
 6155  007566' 472074  000020
 6156  007567' 015100  000000                  MWORD   <CONT,SOA,OR,A=7,B=7,D=7>               ; 151
 6157  007570' 437073  400340
 6158  007571' 015210  000000                  MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>        ; 152 - 200000,,0
 6159  007572' 432074  000020
 6160  007573' 015310  000000                  MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>      ; 153 - comp
 6161  007574' 472074  000020
 6162  007575' 015400  000000                  MWORD   <CONT,SOA,OR,A=7,B=7,D=7>               ; 154
 6163  007576' 437073  400340
 6164  007577' 015510  000000                  MWORD   <CJS,J=1000,SOA,A=7,B=10,OR,D=2>        ; 155 - 400000,,0
 6165  007600' 432074  000020
 6166  007601' 015610  000000                  MWORD   <CJS,J=1000,SOA,A=7,B=10,XNOR,D=2>      ; 156 - comp
 6167  007602' 472074  000020
 6168  007603' 015701  570000                  MWORD   <JMAP,J=157,BAD>                        ; 157 - done - exit
 6169  007604' 000000  000041
 6170
 6171                                  ;       Error address
 6172
 6173  007605' 016001  600000                  MWORD   <ADDR=160,JMAP,J=160,BAD>               ; 160 - error - exit
 6174  007606' 000000  000041
 6175
 6176                                  ;       Read R10 to get correct data
 6177                                  ;       Read R11 to get actual data
 6178                                  ;       Read R12 to get part of LS addr
 6179                                  ;       Read R13 to get rest of LS addr
 6180
 6181  007607' 020002  042004                  MWORD   <ADDR=200,JMAP,J=204,SOA,A=10,D=1,OENA,SELE,MGC=4>
 6182  007610' 401100  005040
 6183  007611' 020102  042004                  MWORD   <JMAP,J=204,SOA,A=11,D=1,OENA,SELE,MGC=4>
 6184  007612' 401110  005040
 6185  007613' 020202  042004                  MWORD   <JMAP,J=204,SOA,A=12,D=1,OENA,SELE,MGC=4>
 6186  007614' 401120  005040
 6187  007615' 020302  042004                  MWORD   <JMAP,J=204,SOA,A=13,D=1,OENA,SELE,MGC=4>
 6188  007616' 401130  005040
 6189  007617' 020402  040000                  MWORD   <JMAP,J=204>
 6190  007620' 000000  000040
 6191
 6192                                  ;       Write/read each page
 6193
 6194  007621' 100020  000000                  MWORD   <ADDR=1000,CJS,J=2000,SDO,OR,B=12,D=1,MGC=0,LSAD> ; 1000 - 0
 6195  007622' 731005  200020
 6196  007623' 100120  000040                  MWORD   <CJS,J=2000,SDO,OR,B=12,D=1,MGC=40,LSAD>       ; 1001 - 1
 6197  007624' 731005  200020
 6198  007625' 100220  000100                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=100,LSAD>        ; 1002 - 2
 6199  007626' 731005  200020
 6200  007627' 100320  000140                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=140,LSAD>        ; 1003 - 3
 6201  007630' 731005  200020
 6202  007631' 100420  000200                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=200,LSAD>        ; 1004 - 4
 6203  007632' 731005  200020
 6204  007633' 100520  000240                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=240,LSAD>        ; 1005 - 5
 6205  007634' 731005  200020
 6206  007635' 100620  000300                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=300,LSAD>        ; 1006 - 6
 6207  007636' 731005  200020
 6208  007637' 100720  000340                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=340,LSAD>        ; 1007 - 7
 6209  007640' 731005  200020
 6210  007641' 101020  000400                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=400,LSAD>        ; 1010 - 10
 6211  007642' 731005  200020
 6212  007643' 101120  000440                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=440,LSAD>        ; 1011 - 11
 6213  007644' 731005  200020
 6214  007645' 101220  000500                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=500,LSAD>        ; 1012 - 12
 6215  007646' 731005  200020
 6216  007647' 101320  000540                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=540,LSAD>        ; 1013 - 13
 6217  007650' 731005  200020
 6218  007651' 101420  000600                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=600,LSAD>        ; 1014 - 14
 6219  007652' 731005  200020
 6220  007653' 101520  000640                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=640,LSAD>        ; 1015 - 15
 6221  007654' 731005  200020
 6222  007655' 101620  000700                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=700,LSAD>        ; 1016 - 16
 6223  007656' 731005  200020
 6224  007657' 101720  000740                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=740,LSAD>        ; 1017 - 17
 6225  007660' 731005  200020
 6226  007661' 102020  001000                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=1000,LSAD>       ; 1020 - 20
 6227  007662' 731005  200020
 6228  007663' 102120  001040                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=1040,LSAD>       ; 1021 - 21
 6229  007664' 731005  200020
 6230  007665' 102220  001100                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=1100,LSAD>       ; 1022 - 22
 6231  007666' 731005  200020
 6232  007667' 102320  001140                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=1140,LSAD>       ; 1023 - 23
 6233  007670' 731005  200020
 6234  007671' 102420  001200                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=1200,LSAD>       ; 1024 - 24
 6235  007672' 731005  200020
 6236  007673' 102520  001240                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=1240,LSAD>       ; 1025 - 25
 6237  007674' 731005  200020
 6238  007675' 102620  001300                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=1300,LSAD>       ; 1026 - 26
 6239  007676' 731005  200020
 6240  007677' 102720  001340                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=1340,LSAD>       ; 1027 - 27
 6241  007700' 731005  200020
 6242  007701' 103020  001400                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=1400,LSAD>       ; 1030 - 30
 6243  007702' 731005  200020
 6244  007703' 103120  001440                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=1440,LSAD>       ; 1031 - 31
 6245  007704' 731005  200020
 6246  007705' 103220  001500                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=1500,LSAD>       ; 1032 - 32
 6247  007706' 731005  200020
 6248  007707' 103320  001540                  MWORD   <CJS,J=2000,SD0,OR,B=12,D=1,MGC=1540,LSAD>       ; 1033 - 33
 6249  007710' 731005  200020
 6250  007711' 103420  001600                  MWORD   <CJS,J=2000,SD0,OR,B=12.D=1,MGC=1600,LSAD>       ; 1034 - 34
 6251  007712' 731005  200020
```

```
007713' 103520  001640          MWORD   <CJS,J=2000,SDO,OR,B=12,D=1,MGC=1640,LSAD>         ; 1035 - 35
007714' 731005  200020
007715' 103620  001700          MWORD   <CJS,J=2000,SDO,OR,B=12,D=1,MGC=1700,LSAD>         ; 1036 - 36
007716' 731005  200020
007717' 103720  001740          MWORD   <CJS,J=2000,SDO,OR,B=12,D=1,MGC=1740,LSAD>         ; 1037 - 37
007720' 731005  200020
007721' 104000  000000          MWORD   <CRTN>                                             ; 1040 - exit
007722' 000000  000240

                        ;       Write/read the LS address locations

007723' 200000  002000          MWORD   <ADDR=2000,CONT,MGC=0,SOA,OR,A=10,OENA,D=1,RAM,LDLM> ; 2000 - 0
007724' 431300  230340
007725' 200100  000000          MWORD   <CONT,MGC=0,SDO,OR,D=2,B=11,RAM,RDLM>              ; 2001
007726' 732204  620340
007727' 200220  040000          MWORD   <CJP,J=2004,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>       ; 2002
007730' 161504  420060
007731' 200301  600000          MWORD   <JMAP,J=160,SDO,SKCN,MGC=0,OR,B=12,D=2>            ; 2003
007732' 732005  240040

007733' 200400  002001          MWORD   <CONT,MGC=1,SOA,OR,A=10,OENA,D=1,RAM,LDLM>         ; 2004 - 1
007734' 431300  230340
007735' 200500  000001          MWORD   <CONT,MGC=1,SDO,OR,D=2,B=11,RAM,RDLM>              ; 2005
007736' 732204  620340
007737' 200620  100000          MWORD   <CJP,J=2010,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>       ; 2006
007740' 161504  420060
007741' 200701  600001          MWORD   <JMAP,J=160,SDO,SKCN,MGC=1,OR,B=12,D=2>            ; 2007
007742' 732005  240040

007743' 201000  002002          MWORD   <CONT,MGC=2,SOA,OR,A=10,OENA,D=1,RAM,LDLM>         ; 2010 - 2
007744' 431300  230340
007745' 201100  000002          MWORD   <CONT,MGC=2,SDO,OR,D=2,B=11,RAM,RDLM>              ; 2011
007746' 732204  620340
007747' 201220  140000          MWORD   <CJP,J=2014,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>       ; 2012
007750' 161504  420060
007751' 201301  600002          MWORD   <JMAP,J=160,SDO,SKCN,MGC=2,OR,B=12,D=2>            ; 2013
007752' 732005  240040

007753' 201400  002003          MWORD   <CONT,MGC=3,SOA,OR,A=10,OENA,D=1,RAM,LDLM>         ; 2014 - 3
007754' 431300  230340
007755' 201500  000003          MWORD   <CONT,MGC=3,SDO,OR,D=2,B=11,RAM,RDLM>              ; 2015
007756' 732204  620340
007757' 201620  200000          MWORD   <CJP,J=2020,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>       ; 2016
007760' 161504  420060
007761' 201701  600003          MWORD   <JMAP,J=160,SDO,SKCN,MGC=3,OR,B=12,D=2>            ; 2017
007762' 732005  240040

007763' 202000  002004          MWORD   <CONT,MGC=4,SOA,OR,A=10,OENA,D=1,RAM,LDLM>         ; 2020 - 4
007764' 431300  230340
007765' 202100  000004          MWORD   <CONT,MGC=4,SDO,OR,D=2,B=11,RAM,RDLM>              ; 2021
007766' 732204  620340
007767' 202220  240000          MWORD   <CJP,J=2024,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>       ; 2022
007770' 161504  420060
007771' 202301  600004          MWORD   <JMAP,J=160,SDO,SKCN,MGC=4,OR,B=12,D=2>            ; 2023
007772' 732005  240040
```

```

007773' 202400  002005          MWORD   <CONT,MGC=5,SOA,OR,A=10,OENA,D=1,RAM,LDLM>      ; 2024 - 5
007774' 431300  230340
007775' 202500  000005          MWORD   <CONT,MGC=5,SDO,OR,D=2,B=11,RAM,RDLM>           ; 2025
007776' 732204  620340
007777' 202620  300000          MWORD   <CJP,J=2030,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>    ; 2026
010000' 161504  420060
010001' 202701  600005          MWORD   <JMAP,J=160,SDO,SKCN,MGC=5,OR,B=12,D=2>         ; 2027
010002' 732005  240040

010003' 203000  002006          MWORD   <CONT,MGC=6,SOA,OR,A=10,OENA,D=1,RAM,LDLM>      ; 2030 - 6
010004' 431300  230340
010005' 203100  000006          MWORD   <CONT,MGC=6,SDO,OR,D=2,B=11,RAM,RDLM>           ; 2031
010006' 732204  620340
010007' 203220  340000          MWORD   <CJP,J=2034,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>    ; 2032
010010' 161504  420060
010011' 203301  600006          MWORD   <JMAP,J=160,SDO,SKCN,MGC=6,OR,B=12,D=2>         ; 2033
010012' 732005  240040

010013' 203400  002007          MWORD   <CONT,MGC=7,SOA,OR,A=10,OENA,D=1,RAM,LDLM>      ; 2034 - 7
010014' 431300  230340
010015' 203500  000007          MWORD   <CONT,MGC=7,SDO,OR,D=2,B=11,RAM,RDLM>           ; 2035
010016' 732204  620340
010017' 203620  400000          MWORD   <CJP,J=2040,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>    ; 2036
010020' 161504  420060
010021' 203701  600007          MWORD   <JMAP,J=160,SDO,SKCN,MGC=7,OR,B=12,D=2>         ; 2037
010022' 732005  240040

010023' 204000  002010          MWORD   <CONT,MGC=10,SOA,OR,A=10,OENA,D=1,RAM,LDLM>     ; 2040 - 10
010024' 431300  230340
010025' 204100  000010          MWORD   <CONT,MGC=10,SDO,OR,D=2,B=11,RAM,RDLM>          ; 2041
010026' 732204  620340
010027' 204220  440000          MWORD   <CJP,J=2044,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>    ; 2042
010030' 161504  420060
010031' 204301  600010          MWORD   <JMAP,J=160,SDO,SKCN,MGC=10,OR,B=12,D=2>        ; 2043
010032' 732005  240040

010033' 204400  002011          MWORD   <CONT,MGC=11,SOA,OR,A=10,OENA,D=1,RAM,LDLM>     ; 2044 - 11
010034' 431300  230340
010035' 204500  000011          MWORD   <CONT,MGC=11,SDO,OR,D=2,B=11,RAM,RDLM>          ; 2045
010036' 732204  620340
010037' 204620  500000          MWORD   <CJP,J=2050,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>    ; 2046
010040' 161504  420060
010041' 204701  600011          MWORD   <JMAP,J=160,SDO,SKCN,MGC=11,OR,B=12,D=2>        ; 2047
010042' 732005  240040

010043' 205000  002012          MWORD   <CONT,MGC=12,SOA,OR,A=10,OENA,D=1,RAM,LDLM>     ; 2050 - 12
010044' 431300  230340
010045' 205100  000012          MWORD   <CONT,MGC=12,SDO,OR,D=2,B=11,RAM,RDLM>          ; 2051
010046' 732204  620340
010047' 205220  540000          MWORD   <CJP,J=2054,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>    ; 2052
010050' 161504  420060
010051' 205301  600012          MWORD   <JMAP,J=160,SDO,SKCN,MGC=12,OR,B=12,D=2>        ; 2053
010052' 732005  240040

```

```
 6362  010053' 205400  002013                MWORD   <CONT,MGC=13,SOA,OR,A=10,OENA,D=1,RAM,LDLM>      ; 2054 - 13
 6363  010054' 431300  230340
 6364  010055' 205500  000013                MWORD   <CONT,MGC=13,SDO,OR,D=2,B=11,RAM,RDLM>           ; 2055
 6365  010056' 732204  620340
 6366  010057' 205620  600000                MWORD   <CJP,J=2060,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>     ; 2056
 6367  010060' 161504  420060
 6368  010061' 205701  600013                MWORD   <JMAP,J=160,SDO,SKCN,MGC=13,OR,B=12,D=2>         ; 2057
 6369  010062' 732005  240040
 6370
 6371  010063' 206000  002014                MWORD   <CONT,MGC=14,SOA,OR,A=10,OENA,D=1,RAM,LDLM>      ; 2060 - 14
 6372  010064' 431300  230340
 6373  010065' 206100  000014                MWORD   <CONT,MGC=14,SDO,OR,D=2,B=11,RAM,RDLM>           ; 2061
 6374  010066' 732204  620340
 6375  010067' 206220  640000                MWORD   <CJP,J=2064,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>     ; 2062
 6376  010070' 161504  420060
 6377  010071' 206301  600014                MWORD   <JMAP,J=160,SDO,SKCN,MGC=14,OR,B=12,D=2>         ; 2063
 6378  010072' 732005  240040
 6379
 6380  010073' 206400  002015                MWORD   <CONT,MGC=15,SOA,OR,A=10,OENA,D=1,RAM,LDLM>      ; 2064 - 15
 6381  010074' 431300  230340
 6382  010075' 206500  000015                MWORD   <CONT,MGC=15,SDO,OR,D=2,B=11,RAM,RDLM>           ; 2065
 6383  010076' 732204  620340
 6384  010077' 206620  700000                MWORD   <CJP,J=2070,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>     ; 2066
 6385  010100' 161504  420060
 6386  010101' 206701  600015                MWORD   <JMAP,J=160,SDO,SKCN,MGC=15,OR,B=12,D=2>         ; 2067
 6387  010102' 732005  240040
 6388
 6389  010103' 207000  002016                MWORD   <CONT,MGC=16,SOA,OR,A=10,OENA,D=1,RAM,LDLM>      ; 2070 - 16
 6390  010104' 431300  230340
 6391  010105' 207100  000016                MWORD   <CONT,MGC=16,SDO,OR,D=2,B=11,RAM,RDLM>           ; 2071
 6392  010106' 732204  620340
 6393  010107' 207220  740000                MWORD   <CJP,J=2074,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>     ; 2072
 6394  010110' 161504  420060
 6395  010111' 207301  600016                MWORD   <JMAP,J=160,SDO,SKCN,MGC=16,OR,B=12,D=2>         ; 2073
 6396  010112' 732005  240040
 6397
 6398  010113' 207400  002017                MWORD   <CONT,MGC=17,SOA,OR,A=10,OENA,D=1,RAM,LDLM>      ; 2074 - 17
 6399  010114' 431300  230340
 6400  010115' 207500  000017                MWORD   <CONT,MGC=17,SDO,OR,D=2,B=11,RAM,RDLM>           ; 2075
 6401  010116' 732204  620340
 6402  010117' 207621  000000                MWORD   <CJP,J=2100,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>     ; 2076
 6403  010120' 161504  420060
 6404  010121' 207701  600017                MWORD   <JMAP,J=160,SDO,SKCN,MGC=17,OR,B=12,D=2>         ; 2077
 6405  010122' 732005  240040
 6406
 6407  010123' 210000  002020                MWORD   <CONT,MGC=20,SOA,OR,A=10,OENA,D=1,RAM,LDLM>      ; 2100 - 20
 6408  010124' 431300  230340
 6409  010125' 210100  000020                MWORD   <CONT,MGC=20,SDO,OR,D=2,B=11,RAM,RDLM>           ; 2101
 6410  010126' 732204  620340
 6411  010127' 210221  040000                MWORD   <CJP,J=2104,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>     ; 2102
 6412  010130' 161504  420060
 6413  010131' 210301  600020                MWORD   <JMAP,J=160,SDO,SKCN,MGC=20,OR,B=12,D=2>         ; 2103
 6414  010132' 732005  240040
 6415
 6416  010133' 210400  002021                MWORD   <CONT,MGC=21,SOA,OR,A=10,OENA,D=1,RAM,LDLM>      ; 2104 - 21
 6417  010134' 431300  230340
 6418  010135' 210500  000021          MWORD  <CONT,MGC=21,SD0,OR,D=2,B=11,RAM,RDLM>          ; 2105
 6419  010136' 732204  620340
 6420  010137' 210621  100000          MWORD  <CJP,J=2110,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>    ; 2106
 6421  010140' 161504  420060
 6422  010141' 210701  600021          MWORD  <JMAP,J=160,SD0,SKCN,MGC=21,OR,B=12,D=2>        ; 2107
 6423  010142' 732005  240040
 6424
 6425  010143' 211000  002022          MWORD  <CONT,MGC=22,S0A,OR,A=10,0ENA,D=1,RAM,LDLM>     ; 2110 - 22
 6426  010144' 431300  230340
 6427  010145' 211100  000022          MWORD  <CONT,MGC=22,SD0,OR,D=2,B=11,RAM,RDLM>          ; 2111
 6428  010146' 732204  620340
 6429  010147' 211221  140000          MWORD  <CJP,J=2114,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>    ; 2112
 6430  010150' 161504  420060
 6431  010151' 211301  600022          MWORD  <JMAP,J=160,SD0,SKCN,MGC=22,OR,B=12,D=2>        ; 2113
 6432  010152' 732005  240040
 6433
 6434  010153' 211400  002023          MWORD  <CONT,MGC=23,S0A,OR,A=10,0ENA,D=1,RAM,LDLM>     ; 2114 - 23
 6435  010154' 431300  230340
 6436  010155' 211500  000023          MWORD  <CONT,MGC=23,SD0,OR,D=2,B=11,RAM,RDLM>          ; 2115
 6437  010156' 732204  620340
 6438  010157' 211621  200000          MWORD  <CJP,J=2120,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>    ; 2116
 6439  010160' 161504  420060
 6440  010161' 211701  600023          MWORD  <JMAP,J=160,SD0,SKCN,MGC=23,OR,B=12,D=2>        ; 2117
 6441  010162' 732005  240040
 6442
 6443  010163' 212000  002024          MWORD  <CONT,MGC=24,S0A,OR,A=10,0ENA,D=1,RAM,LDLM>     ; 2120 - 24
 6444  010164' 431300  230340
 6445  010165' 212100  000024          MWORD  <CONT,MGC=24,SD0,OR,D=2,B=11,RAM,RDLM>          ; 2121
 6446  010166' 732204  620340
 6447  010167' 212221  240000          MWORD  <CJP,J=2124,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>    ; 2122
 6448  010170' 161504  420060
 6449  010171' 212301  600024          MWORD  <JMAP,J=160,SD0,SKCN,MGC=24,OR,B=12,D=2>        ; 2123
 6450  010172' 732005  240040
 6451
 6452  010173' 212400  002025          MWORD  <CONT,MGC=25,S0A,OR,A=10,0ENA,D=1,RAM,LDLM>     ; 2124 - 25
 6453  010174' 431300  230340
 6454  010175' 212500  000025          MWORD  <CONT,MGC=25,SD0,OR,D=2,B=11,RAM,RDLM>          ; 2125
 6455  010176' 732204  620340
 6456  010177' 212621  300000          MWORD  <CJP,J=2130,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>    ; 2126
 6457  010200' 161504  420060
 6458  010201' 212701  600025          MWORD  <JMAP,J=160,SD0,SKCN,MGC=25,OR,B=12,D=2>        ; 2127
 6459  010202' 732005  240040
 6460
 6461  010203' 213000  002026          MWORD  <CONT,MGC=26,S0A,OR,A=10,0ENA,D=1,RAM,LDLM>     ; 2130 - 26
 6462  010204' 431300  230340
 6463  010205' 213100  000026          MWORD  <CONT,MGC=26,SD0,OR,D=2,B=11,RAM,RDLM>          ; 2131
 6464  010206' 732204  620340
 6465  010207' 213221  340000          MWORD  <CJP,J=2134,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>    ; 2132
 6466  010210' 161504  420060
 6467  010211' 213301  600026          MWORD  <JMAP,J=160,SD0,SKCN,MGC=26,OR,B=12,D=2>        ; 2133
 6468  010212' 732005  240040
 6469
 6470  010213' 213400  002027          MWORD  <CONT,MGC=27,S0A,OR,A=10,0ENA,D=1,RAM,LDLM>     ; 2134 - 27
 6471  010214' 431300  230340
```

```
6472  010215' 213500  000027          MWORD   <CONT,MGC=27,SDO,OR,D=2,B=11,RAM,RDLM>              ; 2135
6473  010216' 732204  620340
6474  010217' 213621  400000          MWORD   <CJP,J=2140,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>        ; 2136
6475  010220' 161504  420060
6476  010221' 213701  600027          MWORD   <JMAP,J=160,SDO,SKCN,MGC=27,OR,B=12,D=2>            ; 2137
6477  010222' 732005  240040
6478
6479  010223' 214000  002030          MWORD   <CONT,MGC=30,SOA,OR,A=10,OENA,D=1,RAM,LDLM>         ; 2140 - 30
6480  010224' 431300  230340
6481  010225' 214100  000030          MWORD   <CONT,MGC=30,SDO,OR,D=2,B=11,RAM,RDLM>              ; 2141
6482  010226' 732204  620340
6483  010227' 214221  440000          MWORD   <CJP,J=2144,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>        ; 2142
6484  010230' 161504  420060
6485  010231' 214301  600030          MWORD   <JMAP,J=160,SDO,SKCN,MGC=30,OR,B=12,D=2>            ; 2143
6486  010232' 732005  240040
6487
6488  010233' 214400  002031          MWORD   <CONT,MGC=31,SOA,OR,A=10,OENA,D=1,RAM,LDLM>         ; 2144 - 31
6489  010234' 431300  230340
6490  010235' 214500  000031          MWORD   <CONT,MGC=31,SDO,OR,D=2,B=11,RAM,RDLM>              ; 2145
6491  010236' 732204  620340
6492  010237' 214621  500000          MWORD   <CJP,J=2150,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>        ; 2146
6493  010240' 161504  420060
6494  010241' 214701  600031          MWORD   <JMAP,J=160,SDO,SKCN,MGC=31,OR,B=12,D=2>            ; 2147
6495  010242' 732005  240040
6496
6497  010243' 215000  002032          MWORD   <CONT,MGC=32,SOA,OR,A=10,OENA,D=1,RAM,LDLM>         ; 2150 - 32
6498  010244' 431300  230340
6499  010245' 215100  000032          MWORD   <CONT,MGC=32,SDO,OR,D=2,B=11,RAM,RDLM>              ; 2151
6500  010246' 732204  620340
6501  010247' 215221  540000          MWORD   <CJP,J=2154,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>        ; 2152
6502  010250' 161504  420060
6503  010251' 215301  600032          MWORD   <JMAP,J=160,SDO,SKCN,MGC=32,OR,B=12,D=2>            ; 2153
6504  010252' 732005  240040
6505
6506  010253' 215400  002033          MWORD   <CONT,MGC=33,SOA,OR,A=10,OENA,D=1,RAM,LDLM>         ; 2154 - 33
6507  010254' 431300  230340
6508  010255' 215500  000033          MWORD   <CONT,MGC=33,SDO,OR,D=2,B=11,RAM,RDLM>              ; 2155
6509  010256' 732204  620340
6510  010257' 215621  600000          MWORD   <CJP,J=2160,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>        ; 2156
6511  010260' 161504  420060
6512  010261' 215701  600033          MWORD   <JMAP,J=160,SDO,SKCN,MGC=33,OR,B=12,D=2>            ; 2157
6513  010262' 732005  240040
6514
6515  010263' 216000  002034          MWORD   <CONT,MGC=34,SOA,OR,A=10,OENA,D=1,RAM,LDLM>         ; 2160 - 34
6516  010264' 431300  230340
6517  010265' 216100  000034          MWORD   <CONT,MGC=34,SDO,OR,D=2,B=11,RAM,RDLM>              ; 2161
6518  010266' 732204  620340
6519  010267' 216221  640000          MWORD   <CJP,J=2164,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>        ; 2162
6520  010270' 161504  420060
6521  010271' 216301  600034          MWORD   <JMAP,J=160,SDO,SKCN,MGC=34,OR,B=12,D=2>            ; 2163
6522  010272' 732005  240040
6523
6524  010273' 216400  002035          MWORD   <CONT,MGC=35,SOA,OR,A=10,OENA,D=1,RAM,LDLM>         ; 2164 - 35
6525  010274' 431300  230340
6526  010275' 216500  000035          MWORD   <CONT,MGC=35,SDO,OR,D=2,B=11,RAM,RDLM>              ; 2165
```

```
  6527  010276' 732204  620340
  6528  010277' 216621  700000              MWORD   <CJP,J=2170,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>     ; 2166
  6529  010300' 161504  420060
  6530  010301' 216701  600035              MWORD   <JMAP,J=160,SDO,SKCN,MGC=35,OR,B=12,D=2>         ; 2167
  6531  010302' 732005  240040
  6532
  6533  010303' 217000  002036              MWORD   <CONT,MGC=36,SOA,OR,A=10,OENA,D=1,RAM,LDLM>      ; 2170 - 36
  6534  010304' 431300  230340
  6535  010305' 217100  000036              MWORD   <CONT,MGC=36,SDO,OR,D=2,B=11,RAM,RDLM>           ; 2171
  6536  010306' 732204  620340
  6537  010307' 217221  740000              MWORD   <CJP,J=2174,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>     ; 2172
  6538  010310' 161504  420060
  6539  010311' 217301  600036              MWORD   <JMAP,J=160,SDO,SKCN,MGC=36,OR,B=12,D=2>         ; 2173
  6540  010312' 732005  240040
  6541
  6542  010313' 217400  002037              MWORD   <CONT,MGC=37,SOA,OR,A=10,OENA,D=1,RAM,LDLM>      ; 2174 - 37
  6543  010314' 431300  230340
  6544  010315' 217500  000037              MWORD   <CONT,MGC=37,SDO,OR,D=2,B=11,RAM,RDLM>           ; 2175
  6545  010316' 732204  620340
  6546  010317' 217622  000000              MWORD   <CJP,J=2200,SAB,XOR,A=10,B=11,D=1,CENA,CCFZ>     ; 2176
  6547  010320' 161504  420060
  6548  010321' 217701  600037              MWORD   <JMAP,J=160,SDO,SKCN,MGC=37,OR,B=12,D=2>         ; 2177
  6549  010322' 732005  240040
  6550
  6551  010323' 220000  000000              MWORD   <CRTN>                                           ; 2200
  6552  010324' 000000  000240
  6553  010325' 777777  777777              -1
```

```
                                  ;#**********************************************************************
                                  ;* TEST 36 - Local Store Address Test
                                  ;
                                  ; Description:  This test writes an address pattern to each local
                                  ;               storage word, and reads it back and verifies  the
                                  ;               data read.
                                  ;
                                  ; Procedure:    Port Clear
                                  ;               Load microcode to write address pattern to each
                                  ;                 local storage address
                                  ;               Start up microcode (this writes each address)
                                  ;               Stop microcode
                                  ;
                                  ;               Set start address
                                  ;               Single step 2 times
                                  ;               Read EBUF and verify data
                                  ;
                                  ;               Repeat above 3 steps for addresses 0,1,..1777
                                  ;
                                  ; Failure:      ---
                                  ;#**********************************************************************

                                  ; Test data

010326' 254 00 0 00 010334'       TSTU36: JRST    TG36                            ; go start test
010327' 200403  000036                    MPROC!NDMP!ZMPROC!36                    ; test mask
010330' 000000  013073'                   0,,[ASCIZ ^Local Store Address Test^]
010331' 013100' 013102'                   [EXP M14,MLAST!M15],,[EXP M2,M13,MLAST!M11]
010332' 000000  010521'                   TSTU37                                  ; failure test table
010333' 777777  777777                    -1                                      ;  ...

                                  ; Start test

010334' 201 00 0 00 000000'       TG36:   MOVEI   Z8                              ; get address of module start
010335' 260 17 0 00 007147*               GO      TRACE                           ; handle trace output

                                  ; Initialization

010336' 400 15 0 00 000000        TL36:   SETZ    ERFLG,                          ; clear error flag
010337' 260 17 0 00 007154*               GO      IPACLR                          ; clear port
010340' 402 00 0 00 007155*               SETZM   TSTSUB                          ; initialize subtest number
010341' 201 06 0 00 010353'               MOVEI   6,TS36                          ; get sstep table address
```

```

                                        ; Loop on single step table entries

010342' 260 17 0 00 007046*             TA36:   GO      BEXEC                   ; execute table entry
010343' 254 00 0 00 010352'                     JRST    TX36                    ; end of sstep table
010344' 254 00 0 00 010342'                     JRST    TA36                    ; keep looping after call
                                                GO      [MOVE   AEBUF           ; EBUF data may be incorrect - do
                                                         AND    [777400,,1777]  ;   compare after masking out the
                                                         CAME   CEBUF           ;   middle 16 indeterminate bits
                                                         SETO   ERFLG,          ; incorrect - set error flag
010345' 260 17 0 00 013106'                              RTN]

                                        ; Handle error printouts and scope looping

010346' 027 00 0 00 010470'                     SCOPER  MA36                    ; print error message
010347' 254 00 0 00 010336'                     JRST    TL36                    ; loop on error
010350' 254 00 0 00 010352'                     JRST    TX36                    ; altmode exit
010351' 322 15 0 00 010342'                     JUMPE   ERFLG,TA36              ; do next sstep table entry

                                        ; End of test

010352' 263 17 0 00 000000              TX36:   RTN                             ; return

                                        ; Sstep table, with EBUF data, as:  (CMD,#Steps,Start addr,End Addr,EBUF data)

010353' 300000  010361'                 TS36:   ATABLE  (SSCALL,TS36LD)         ; load microcode/init test paramters
010354' 400000  010457'                 TS36L:  ATABLE  (SSCHK,TS36ST)          ; set up single step parameters
010355' 100200  002000                          ATABLE  (SSSTRT,2,0,2000,0)     ; do 3 single steps/check results
010356' 000000  000000
010357' 500000  010354'                         ATABLE  (SSJRST,TS36L)          ; do next address
010360' 600000  000000                          ATABLE  (SSLAST)                ; end of table

                                        ; Special routine - load microcode to write address pattern to local storage

010361' 316 16 0 00 007103*             TS36LD: CAMN    MBCN,PORTNI             ; NI port?
010362' 476 00 0 00 007104*                     SETOM   TSLOD1                  ; yes - must reload microcode each time
010363' 316 16 0 00 007105*                     CAMN    MBCN,PORTCI             ; CI port?
010364' 476 00 0 00 007106*                     SETOM   TSLOD2                  ; yes - must reload microcode each time
010365' 201 01 0 00 010503'                     MOVEI   1,T36MA                 ; get address of ucode
010366' 260 17 0 00 007151*                     GO      TLOAD                   ; load it
010367' 255 00 0 00 000000                       JFCL                           ; error - ignore
010370' 402 00 0 00 001033*                     SETZM   CADDR                   ; clear CRAM address
010371' 260 17 0 00 000000*                     GO      DRCRAM                  ; read CRAM location
010372' 120 00 0 00 001037*                     DMOVE   CWORDL                  ; get CRAM data
010373' 124 00 0 00 010453'                     DMOVEM  TS36W0                  ; save this word
010374' 350 00 0 00 010370*                     AOS     CADDR                   ; point to next address
010375' 260 17 0 00 010371*                     GO      DRCRAM                  ; read CRAM location
010376' 120 00 0 00 010372*                     DMOVE   CWORDL                  ; get CRAM data
010377' 124 00 0 00 010455'                     DMOVEM  TS36W1                  ; save this word
010400' 261 17 0 00 001044*                     PUT     PARFLG                  ; save state of parity flag
010401' 476 00 0 00 010400*                     SETOM   PARFLG                  ; ensure it is set
```

```

                                     ; Now load CRAM locations 2-3777 while inserting the local storage
                                     ; address in the magic number field.

010402' 350 02 0 00 010374*         TS36L0: AOS     2,CADDR                 ; point to next address
010403' 303 02 0 00 003777                  CAILE   2,3777                  ; done yet?
010404' 254 00 0 00 010420'                 JRST    TS36L1                  ; yes - do next part
010405' 120 00 0 00 010453'                 DMOVE   TS36W0                  ; get first word
010406' 124 00 0 00 010376*                 DMOVEM  CWORDL                  ; save it
010407' 242 02 0 00 777777                  LSH     2,-1                    ; make 1-1777
010410' 137 02 0 00 013113'                 DPB     2,[POINT 10,CWORDL,29]  ; save MGC #
010411' 260 17 0 00 001043*                 GO      DWCRAM                  ; write to CRAM
010412' 120 00 0 00 010455'                 DMOVE   TS36W1                  ; get second word
010413' 124 00 0 00 010406*                 DMOVEM  CWORDL                  ; save it
010414' 137 02 0 00 013113'                 DPB     2,[POINT 10,CWORDL,29]  ; save MGC #
010415' 350 00 0 00 010402*                 AOS     CADDR                   ; point to next address
010416' 260 17 0 00 010411*                 GO      DWCRAM                  ; write to CRAM
010417' 254 00 0 00 010402'                 JRST    TS36L0                  ; loop till done

                                     ; Now the first 4001 location are set up to write address pattern
                                     ; to local storage locations 0-1023 (decimal).  Now start up the
                                     ; port to accomplish this.

010420' 201 01 0 00 004001          TS36L1: MOVEI   1,4001                  ; get address to start up at
010421' 242 01 0 00 000001                  LSH     1,1                     ; position properly
010422' 260 17 0 00 007175*                 GO      LDRAR                   ; write to RAR register
010423' 201 01 0 00 000010                  MOVEI   1,MPRUN                 ; get CSR start data
010424' 260 17 0 00 007201*                 GO      LDCSR                   ; start up the port
010425' 201 00 0 00 000100                  MOVEI   100                     ; set up to wait for 100 ms
010426' 260 17 0 00 000000*                 GO      ODELAY                  ;   for this to complete
010427' 400 01 0 00 000000                  SETZ    1,                      ; clear CSR data
010430' 260 17 0 00 010424*                 GO      LDCSR                   ; stop the port

                                     ; Now load the CRAM with instructions to read each location

010431' 316 16 0 00 010361*                 CAMN    MBCN,PORTNI             ; NI port?
010432' 476 00 0 00 010362*                 SETOM   TSLOD1                  ; yes - must reload microcode each time
010433' 316 16 0 00 010363*                 CAMN    MBCN,PORTCI             ; CI port?
010434' 476 00 0 00 010364*                 SETOM   TSLOD2                  ; yes - must reload microcode each time
010435' 201 01 0 00 010514'                 MOVEI   1,T36MB                 ; get address of ucode
010436' 260 17 0 00 010366*                 GO      TLOAD                   ; load it
010437' 255 00 0 00 000000                   JFCL                           ; error - ignore
010440' 402 00 0 00 010415*                 SETZM   CADDR                   ; clear CRAM address
010441' 260 17 0 00 010375*                 GO      DRCRAM                  ; read CRAM location
```

```

                                         ; Now load CRAM locations 1-1777 while inserting the local storage
                                         ; address in the magic number field.

010442' 350 01 0 00 010440*        TS36L2: AOS     1,CADDR                     ; point to next address
010443' 303 01 0 00 001777                 CAILE   1,1777                      ; done yet?
010444' 254 00 0 00 010450'                JRST    TS36L3                      ; yes - exit
010445' 137 01 0 00 013113'                DPB     1,[POINT 10,CWORDL,29]      ; save MGC #
010446' 260 17 0 00 010416*                GO      DWCRAM                      ; write to CRAM
010447' 254 00 0 00 010442'                JRST    TS36L2                      ; loop till done

010450' 474 07 0 00 000000         TS36L3: SETO    7,                          ; initialize start address
010451' 262 17 0 00 010401*                GET     PARFLG                      ; restore state of parity flag
010452' 263 17 0 00 000000                 RTN                                 ; return

                                         ; Miscellaneous data

010453'                            TS36W0: BLOCK   2                           ; 1st microword
010455'                            TS36W1: BLOCK   2                           ; 2nd microword

                                         ; Special routine - set up for next local storage address

010457' 350 00 0 00 000007         TS36ST: AOS     7                           ; point to next address
010460' 202 07 0 00 011165'                MOVEM   7,SAVDAT                    ; save address for error printout
010461' 137 07 0 00 013114'                DPB     7,[POINT 10,SAVDAT,9]       ; insert address into left 10 bits
010462' 307 07 0 00 001777                 CAIG    7,1777                      ; done yet?
010463' 350 00 0 17 000000                 AOS     (P)                         ; no - set up RTN+2
010464' 137 07 0 00 013115'                DPB     7,[POINT 12,TS36L+1,23]     ; set up single step start address
010465' 200 00 0 00 011165'                MOVE    SAVDAT                      ; set up correct EBUF data (LS addr
010466' 202 00 0 00 010356'                MOVEM   TS36L+2                     ;   in each side of word)
010467' 263 17 0 00 000000                 RTN                                 ; return

                                         ; Error messages

010470' 260000  010472'            MA36:   CALL!TXALL!MA36PN                   ; print data
010471' 000000000000#                      LAST!CALL!TXALL!BBPNT               ; print sequencer data

010472' 200 01 0 00 007221*        MA36PN: MOVE    1,SCOSW                     ; get switches
010473' 607 01 0 00 000200                 TLNN    1,TXTINH                    ; text inhibit switch set?
010474' 037 00 0 00 013116'                TMSGC   <Wrote address patterns to each local storage location.>
010475' 037 00 0 00 013132'                TMSGC   <Error at >
010476' 200 00 0 00 011165'                MOVE    SAVDAT#                     ; get address
010477' 037 04 0 00 000000                 PNT4                                ; print it
010500' 607 01 0 00 000200                 TLNN    1,TXTINH                    ; text inhibit switch set?
010501' 037 00 0 00 013135'                TMSG    < (Result in EBUF - middle 16 bits indeterminate)>
010502' 263 17 0 00 000000                 RTN
```

```
 6738
 6739                                      ; Microcode to write address pattern to each local storage address
 6740
 6741   010503' 000000  000000            T36MA:  MWORD   <ADDR=0,CONT,SDO,OR,D=2,SKCN,MGC=0>      ; 0-3777
 6742   010504' 732000  240340
 6743   010505' 000100  002000                    MWORD   <CONT,SOA,OR,D=1,OENA,LDLM,MGC=0>
 6744   010506' 431000  230340
 6745   010507' 400040  000000                    MWORD   <ADDR=4000,JMAP,J=4000>                  ; 4000
 6746   010510' 000000  000040
 6747   010511' 400100  000000                    MWORD   <JZ>                                     ; 4001
 6748   010512' 000000  000000
 6749   010513' 777777  777777                    -1
 6750
 6751                                      ; Microcode to read each local storage address and place data in EBUF
 6752
 6753   010514' 000020  000000            T36MB:  MWORD   <ADDR=0,JMAP,J=2000,SDO,OR,D=2,RDLM,MGC=0>        ; 0-1777
 6754   010515' 732000  220040
 6755   010516' 200020  002004                    MWORD   <ADDR=2000,JMAP,J=2000,SOA,OR,D=1,OENA,SELE,MGC=4> ; 2000
 6756   010517' 431000  005040
 6757   010520' 777777  777777                    -1
```

```

                                    ;#****************************************************************
                                    ;* TEST 37 - Local Store RAM Mode Test
                                    ;
                                    ; Description:  Data is written and verified to two local storage
                                    ;               locations alternating between using the RAM  mode
                                    ;               bit and not.
                                    ;
                                    ; Procedure:    Port Clear
                                    ;               Load test microcode
                                    ;               Set start address 0
                                    ;               Single step 5 times - write 0's to LS 0 and 1's
                                    ;                 to LS 1740 (RAM mode bit not set), then read
                                    ;                 LS 0 (with RAM mode bit set) and put data in
                                    ;                 EBUF
                                    ;               Read EBUF and verify data 1's
                                    ;               Single step 3 times - read LS 0 (RAM mode bit
                                    ;                 not set), then read LS 1740 (with RAM mode bit
                                    ;                 set) and put data in EBUF
                                    ;               Read EBUF and verify data 0's
                                    ;
                                    ; Failure:      ---
                                    ;#****************************************************************

                                    ; Test data

010521' 254 00 0 00 010526'         TSTU37: JRST    TG37                    ; go start test
010522' 200403  000037                      MPROC!NDMP!ZMPROC!37            ; test mask
010523' 010560' 013147'                     T37M,,[ASCIZ ^Local Store RAM Mode Test^]
010524' 013155' 013102'                     [EXP M15,MLAST!M14],,[EXP M2,M13,MLAST!M11]
010525' 777777  777777                      -1                              ; failure test table

                                    ; Start test

010526' 201 00 0 00 000000'         TG37:   MOVEI   Z8                      ; get address of module start
010527' 260 17 0 00 010335*                 GO      TRACE                   ; handle trace output
010530' 201 01 0 00 010560'                 MOVEI   1,T37M                  ; set up microcode address
010531' 260 17 0 00 010436*                 GO      TLOAD                   ; load/verify it
010532' 263 17 0 00 000000                  RTN                             ; failed - exit test

                                    ; Initialization

010533' 400 15 0 00 000000          TL37:   SETZ    ERFLG,                  ; clear error flag
010534' 260 17 0 00 010337*                 GO      IPACLR                  ; clear port
010535' 402 00 0 00 010340*                 SETZM   TSTSUB                  ; initialize subtest number
010536' 201 06 0 00 010550'                 MOVEI   6,TS37                  ; get sstep table address
```

```

                                          ; Loop on single step table entries

 010537' 260 17 0 00 010342*      TA37:   GO      BEXEC                  ; execute table entry
 010540' 254 00 0 00 010547'              JRST    TX37                   ; end of sstep table
 010541' 254 00 0 00 010537'              JRST    TA37                   ; keep looping after call
 010542' 474 15 0 00 000000               SETO    ERFLG,                 ; error - EBUF data incorrect

                                          ; Handle error printouts and scope looping

 010543' 027 00 0 00 010555'              SCOPER  MA37                   ; print error message
 010544' 254 00 0 00 010533'              JRST    TL37                   ; loop on error
 010545' 254 00 0 00 010547'              JRST    TX37                   ; altmode exit
 010546' 322 15 0 00 010537'              JUMPE   ERFLG,TA37             ; do next sstep table entry

                                          ; End of test

 010547' 263 17 0 00 000000       TX37:   RTN                            ; return

                                          ; Sstep table, with EBUF data, as:  (CMD,#Steps,Start addr,End Addr,EBUF data)

 010550' 100600  000006           TS37:   ATABLE  (SSSTRT,6,0,6,-1)      ; do 6 single steps/check results
 010551' 777777  777777
 010552' 100400  060012                   ATABLE  (SSSTRT,4,6,12,0)      ; do 4 single steps/check results
 010553' 000000  000000
 010554' 000000  000000                   ATABLE  (SSLAST)               ; end of table

                                          ; Error messages

 010555' 140000  013157'          MA37:   MSG!TXNOT![ASCIZ /RAM Mode bit failed - wrong local storage location read./]
 010556' 140000  013173'                  MSG!TXNOT![ASCIZ /Result in EBUF./]
 010557' 000000000000#                    LAST!CALL!TXALL!BBPNT          ; print sequencer data
```

```

                                ; Microcode:

                                ;       Write 0's to LS 0, 1's to LS 1740

010560' 000000  010000          T37M:   MWORD   <ADDR=0,JMAP,J=1,SOA,AND,D=2,LSAD,MGC=0>; 0
010561' 442000  200040
010562' 000100  002000                  MWORD   <CONT,SOA,OR,D=1,OENA,LDLM>             ; 1
010563' 431000  230340
010564' 000200  001740                  MWORD   <CONT,SOA,AND,D=1,LSAD,MGC=1740>        ; 2
010565' 441000  200340
010566' 000300  003740                  MWORD   <CONT,SOA,XNOR,D=1,OENA,MGC=1740,LDLM>  ; 3
010567' 471000  230340

                                ;       Read LS 0 and verify result

010570' 000400  000000                  MWORD   <CONT,SDO,OR,D=2,RAM,RDLM,MGC=0>        ; 4
010571' 732200  220340
010572' 000500  002004                  MWORD   <CONT,SOA,OR,D=1,OENA,MGC=4,SELE>       ; 5
010573' 431000  005340

                                ;       Read LS 0, then LS 1740, and verify result

010574' 000600  070000                  MWORD   <JMAP,J=7,SOA,AND,D=1,LSAD,MGC=0>       ; 6
010575' 441000  200040
010576' 000700  000000                  MWORD   <CONT,SDO,D=2,MGC=0,RDLM>               ; 7
010577' 702000  220340
010600' 001000  001740                  MWORD   <CONT,SDO,D=2,MGC=1740,RAM,RDLM>        ; 10
010601' 702200  220340
010602' 001100  002004                  MWORD   <CONT,SOA,D=1,OENA,MGC=4,SELE>          ; 11
010603' 401000  005340
010604' 001200  120000                  MWORD   <JMAP,J=12>                             ; 12
010605' 000000  000040
010606' 777777  777777                  -1
```

```

                                          ;#*****************************************************************
                                          ;* Test 40 - Jump Mux Test
                                          ;
                                          ; Description:  Verify that the JMP MUX can select the low  order
                                          ;               8 bits from the data on the MBUS rather than  the
                                          ;               jump field of the microword.
                                          ;
                                          ; Procedure:    Clear Port
                                          ;               Load test microcode
                                          ;
                                          ;               Set start address = 0
                                          ;               Single step once to initialize R0
                                          ;
                                          ;               Set start address = 1
                                          ;               Single step once to select next address
                                          ;               Read LAR and verify address
                                          ;
                                          ;               Repeat the above 3 steps 255 times for MBUS field
                                          ;               2 to 254.
                                          ;
                                          ; Failure:      JMP MUX not selected properly
                                          ;#*****************************************************************

                                          ; Test data

 010607' 254 00 0 00 010614'              TSTU40: JRST    TG40                      ; go start test
 010610' 210403  000040                           MPROC!MBUS!NDMP!ZMPROC!40         ; test mask
 010611' 010656' 013177'                          T40M,,[ASCIZ ^Jump MUX Test^]
 010612' 013202' 013003'                          [EXP MLAST!M5],,[EXP MLAST!M11]
 010613' 777777  777777                           -1                                ; failure test table

                                          ; Start test

 010614' 201 00 0 00 000000'              TG40:   MOVEI   Z8                        ; get address of module start
 010615' 260 17 0 00 010527*                      GO      TRACE                     ; handle trace output
 010616' 201 01 0 00 010656'                      MOVEI   1,T40M                    ; set up microcode address
 010617' 260 17 0 00 010531*                      GO      TLOAD                     ; load/verify it
 010620' 263 17 0 00 000000                       RTN                               ; failed - exit test

                                          ; Initialization

 010621' 400 15 0 00 000000               TL40:   SETZ    ERFLG,                    ; clear error flag
 010622' 260 17 0 00 010534*                      GO      IPACLR                    ; clear port
 010623' 402 00 0 00 010535*                      SETZM   TSTSUB                    ; initialize subtest number
 010624' 201 06 0 00 010636'                      MOVEI   6,TS40                    ; get sstep table address

                                          ; Loop on single step table entries

 010625' 260 17 0 00 000000*              TA40:   GO      SEXEC                     ; execute table entry
 010626' 254 00 0 00 010635'                      JRST    TX40                      ; end of sstep table
 010627' 254 00 0 00 010625'                      JRST    TA40                      ; keep looping after call
 010630' 474 15 0 00 000000                       SETO    ERFLG,                    ; error - end address incorrect
```

```

                                        ; Handle error printouts and scope looping

010631' 027 00 0 00 010654'                     SCOPER  MA40                    ; print error message
010632' 254 00 0 00 010621'                     JRST    TL40                    ; loop on error
010633' 254 00 0 00 010635'                     JRST    TX40                    ; altmode exit
010634' 322 15 0 00 010625'                     JUMPE   ERFLG,TA40              ; do next entry (if no error yet)

                                        ; End of test

010635' 263 17 0 00 000000              TX40:   RTN                             ; return

                                        ; Single step table, as:  (CMD,#Steps,Start addr,End Addr)

010636' 300000  010644'                 TS40:   STABLE  (SSCALL,TS40IN)         ; init end address
010637' 101700  000004                          STABLE  (SSSTRT,^D15,0,4)       ; init 2901 R0
010640' 400000  010647'                 TS40A:  STABLE  (SSCHK,TS40CH)          ; check if done yet
010641' 100100  040000                          STABLE  (SSSTRT,1,4,0)          ; do the single steps
010642' 500000  010640'                         STABLE  (SSJRST,TS40A)          ; loop till done
010643' 000000  000000                          STABLE  (SSLAST)

010644' 402 00 0 00 013422'             TS40IN: SETZM   SAVADD#                 ; init end address
010645' 350 00 0 00 013422'                     AOS     SAVADD
010646' 263 17 0 00 000000                      RTN                             ; return

010647' 350 01 0 00 013422'             TS40CH: AOS     1,SAVADD                ; point to next address
010650' 305 01 0 00 000377                      CAIGE   1,^D255                 ; done yet?
010651' 350 00 0 17 000000                      AOS     (P)                     ; no - set up proper return
010652' 137 01 0 00 013203'                     DPB     1,[POINT 12,TS40A+1,35] ; save end address
010653' 263 17 0 00 000000                      RTN                             ; return

                                        ; Error message

010654' 140000  013204'                 MA40:   MSG!TXNOT![ASCIZ /JMP MUX not selected correctly/]
010655' 000000000000#                           LAST!CALL!TXALL!SSPNT           ; print sequencer data

                                        ; Test microcode

010656' 000000  010000                  T40M:   MWORD   <ADDR=0,JMAP,J=1,SOA,AND,D=2>           ; 0
010657' 442000  000040
010660' 000100  130000                          MWORD   <LDCT,J=13,SOA,PLUS,D=2,CRY>            ; 1
010661' 402000  000700
010662' 000200  020000                          MWORD   <RPCT,J=2,SOA,OR,D=7>                   ; 2
010663' 437000  000220
010664' 000300  000000                          MWORD   <CONT,SOA,OR,B=1,D=2>                   ; 3
010665' 432000  400340
010666' 000403  772000                          MWORD   <JMAP,J=377,SAB,PLUS,A=1,D=3,SKMB,OENA> ; 4
010667' 103010  210040
010670' 777777  777777                          -1
```

```
                                        ;#*********************************************************************
                                        ;* TEST 41 - CRAM Control Register Parity Generators Test
                                        ;
                                        ; Description:  Verify the parity generators which verify good
                                        ;               parity of each word gated out of the CRAM into
                                        ;               the CRAM control register.
                                        ;
                                        ; Procedure:    Port Clear
                                        ;               Load CRAM location 0 with a data pattern, with good
                                        ;                 parity).
                                        ;               Single step once and verify that a CRAM PE does not
                                        ;                 occur.
                                        ;               Load CRAM location 0 with the same data pattern, but
                                        ;                 with bad parity).
                                        ;               Single step once and verify that a CRAM PE does occur.
                                        ;
                                        ;               Repeat for each data pattern.
                                        ;
                                        ; Failure:      ---
                                        ;#*********************************************************************

                                        ; Test data

010671' 254 00 0 00 010677'             TSTU41: JRST    TG41                    ; go start test
010672' 200403  000041                          MPROC!NDMP!ZMPROC!41            ; test mask
010673' 000000  013213'                         0,,[ASCIZ ^CRAM Ctl Reg Parity Generators Test^]
010674' 011324' 011324'                         [EXP MLAST!M12],,[EXP MLAST!M12]
010675' 000000000000#                           TSTE45                          ; failure test table
010676' 777777  777777                          -1                              ;   ...

                                        ; Start test

010677' 201 00 0 00 000000'             TG41:   MOVEI   Z8                      ; get address of module start
010700' 260 17 0 00 010615*                     GO      TRACE                   ; handle trace output
010701' 200 07 0 00 030022                      MOVE    7,RANDBS                ; get base number
010702' 202 07 0 00 013421'                     MOVEM   7,OLDRAN#               ; save it

                                        ; Initialization

010703' 402 00 0 00 010442*                     SETZM   CADDR                   ; clear CRAM address
010704' 400 06 0 00 000000                      SETZ    6,                      ; initialize data pattern number
010705' 202 06 0 00 010623*             TL41:   MOVEM   6,TSTSUB                ; set up test offset

                                        ; Get data pattern

010706' 400 15 0 00 000000                      SETZ    ERFLG,                  ; clear error flag
010707' 260 17 0 00 011007'                     GO      T41RAN                  ; get random data
010710' 202 02 0 00 010413*                     MOVEM   2,CWORDL                ; save left 30 bits
010711' 202 03 0 00 001040*                     MOVEM   3,CWORDR                ; save right 30 hits
```

```

                                       ; Test segment A

010712' 260 17 0 00 010622*     TA41:  GO       IPACLR              ; clear port
010713' 201 01 0 00 000001             MOVEI    1,1                 ; set bit 12 (for left 30 bits)
010714' 260 17 0 00 010422*            GO       LDRAR               ; load RAR with it
010715' 200 02 0 00 010710*            MOVE     2,CWORDL            ; get left 30 bits
010716' 200 03 0 00 010711*            MOVE     3,CWORDR            ; get right 30 bits
010717' 400 01 0 00 000000             SETZ     1,                  ; set up no force parity
010720' 260 17 0 00 000000*            GO       CALPAR              ; calculate good parity
010721' 200 01 0 00 000002             MOVE     1,2                 ; get left 30 bits
010722' 260 17 0 00 000353*            GO       LDCRAM              ; load it
010723' 400 01 0 00 000000             SETZ     1,                  ; clear bit 12 (for right 30 bits)
010724' 260 17 0 00 010714*            GO       LDRAR               ; load address
010725' 200 01 0 00 000003             MOVE     1,3                 ; get right 30 bits
010726' 260 17 0 00 010722*            GO       LDCRAM              ; load it
010727' 201 01 0 00 020010             MOVEI    1,MPRUN!SINCYC      ; get proper CSR data
010730' 260 17 0 00 010430*            GO       LDCSR               ; single step once
010731' 260 17 0 00 001106*            GO       RDCSR               ; read the CSR register
010732' 474 15 0 00 000000             SETO     ERFLG,              ; error
010733' 603 01 0 00 004000             TLNE     1,(CRAMPE)          ; get a CRAM PE?
010734' 474 15 0 00 000000             SETO     ERFLG,              ; yes - set error flag
010735' 027 00 0 00 010774'            SCOPER   MA41                ; print error message
010736' 254 00 0 00 010712'            JRST     TA41                ; loop on error
010737' 254 00 0 00 010773'            JRST     TX41                ; altmode exit
010740' 326 15 0 00 010773'            JUMPN    ERFLG,TX41          ; error yet? - yes - exit test

                                       ; Test segment B

010741' 260 17 0 00 010712*     TB41:  GO       IPACLR              ; clear port
010742' 201 01 0 00 000001             MOVEI    1,1                 ; set bit 12 (for left 30 bits)
010743' 260 17 0 00 010724*            GO       LDRAR               ; load RAR with it
010744' 200 02 0 00 010715*            MOVE     2,CWORDL            ; get left 30 bits
010745' 200 03 0 00 010716*            MOVE     3,CWORDR            ; get right 30 bits
010746' 474 01 0 00 000000             SETO     1,                  ; set up to force bad parity
010747' 260 17 0 00 010720*            GO       CALPAR              ; calculate good parity
010750' 200 01 0 00 000002             MOVE     1,2                 ; get left 30 bits
010751' 260 17 0 00 010726*            GO       LDCRAM              ; load it
010752' 400 01 0 00 000000             SETZ     1,                  ; clear bit 12 (for right 30 bits)
010753' 260 17 0 00 010743*            GO       LDRAR               ; load address
010754' 200 01 0 00 000003             MOVE     1,3                 ; get right 30 bits
010755' 260 17 0 00 010751*            GO       LDCRAM              ; load it
010756' 201 01 0 00 020010             MOVEI    1,MPRUN!SINCYC      ; get proper CSR data
010757' 260 17 0 00 010730*            GO       LDCSR               ; single step once
010760' 260 17 0 00 010731*            GO       RDCSR               ; read the CSR register
010761' 474 15 0 00 000000             SETO     ERFLG,              ; error
010762' 607 01 0 00 004000             TLNN     1,(CRAMPE)          ; get a CRAM PE?
010763' 474 15 0 00 000000             SETO     ERFLG,              ; yes - set error flag
010764' 027 00 0 00 010776'            SCOPER   MB41                ; print error message
010765' 254 00 0 00 010741'            JRST     TB41                ; loop on error
010766' 254 00 0 00 010773'            JRST     TX41                ; altmode exit
010767' 326 15 0 00 010773'            JUMPN    ERFLG,TX41          ; error yet? - yes - exit test
```

```



                                    ; Next data pattern

010770' 350 00 0 00 000006                  AOS     6                       ; set up next test offset
010771' 307 06 0 00 001750                  CAIG    6,^D1000                ; done yet?
010772' 254 00 0 00 010705'                 JRST    TL41                    ; no - loop till done

                                    ; End of test

010773' 263 17 0 00 000000          TX41:   RTN                             ; return

                                    ; Error messages

010774' 040000  013223'             MA41:   TXNOT![ASCIZ /CRAM PE should not have occurred/]
010775' 270000  011000'                     LAST!CALL!TXALL!M41PNT          ; print error data

010776' 040000  013232'             MB41:   TXNOT![ASCIZ /CRAM PE should have occurred/]
010777' 270000  011000'                     LAST!CALL!TXALL!M41PNT          ; print error data

011000' 037 00 0 00 013240'         M41PNT: TMSGC   <CRAM data:  >
011001' 260 17 0 00 010441*                 GO      DRCRAM                  ; read CRAM data
011002' 200 00 0 00 010744*                 MOVE    CWORDL                  ; get left half
011003' 260 17 0 00 011070'                 GO      MPNTC                   ; print it
011004' 200 00 0 00 010745*                 MOVE    CWORDR                  ; get right half
011005' 260 17 0 00 011070'                 GO      MPNTC                   ; print it
011006' 263 17 0 00 000000                  RTN

                                    ; Random number generator

011007' 270 07 0 00 030022          T41RAN: ADD     7,RANDBS                ; modify number in AC7
011010' 241 07 0 00 777774                  ROT     7,-4                    ; to create a new number
011011' 447 07 0 00 013421'                 EQVB    7,OLDRAN                ; put result in AC7 and OLDRAN
011012' 200 02 0 00 000007                  MOVE    2,7                     ; put into AC2
011013' 270 07 0 00 030022                  ADD     7,RANDBS                ; modify number in AC7
011014' 241 07 0 00 777774                  ROT     7,-4                    ; to create a new number
011015' 447 07 0 00 013421'                 EQVB    7,OLDRAN                ; put result in AC7 and OLDRAN
011016' 200 03 0 00 000007                  MOVE    3,7                     ; put into AC3
011017' 621 02 0 00 770000                  TLZ     2,770000                ; clear left 6 bits
011020' 621 03 0 00 770000                  TLZ     3,770000                ; clear left 6 bits
011021' 120 02 0 00 011002*                 DMOVE   2,CWORDL                ; save the data
011022' 263 17 0 00 000000                  RTN

```

```
                                                SUBTTL  Miscellaneous Routines

                                        ;#*********************************************************************
                                        ; MPNT1 - Print CRAM data correct/actual (data in SAVCR1,SAVCR2)
                                        ;#*********************************************************************

011023' 200 01 0 00 010472*             MPNT1:  MOVE    1,SCOSW                         ; get switches
011024' 037 00 0 00 013243'                     TMSGC   <CRAM Loc: >
011025' 200 00 0 00 011164'                     MOVE    SAVCRA                          ; get address
011026' 242 00 0 00 777777                      LSH     -1                              ; position correctly
011027' 037 04 0 00 000000                      PNT4                                    ; print it
011030' 200 00 0 00 011164'                     MOVE    SAVCRA                          ; get address
011031' 602 00 0 00 000001                      TRNE    1                               ; left half?
011032' 037 00 0 00 013246'                     TMSG    <(left)>                        ; yes - print it
011033' 606 00 0 00 000001                      TRNN    1                               ; right half?
011034' 037 00 0 00 013250'                     TMSG    <(right)>                       ; yes - print it
011035' 603 01 0 00 000200                      TLNE    1,TXTINH                        ; inhibit text?
011036' 037 00 0 00 013252'                     TMSGC   <CRAM (C):  >                   ; yes
011037' 607 01 0 00 000200                      TLNN    1,TXTINH                        ; inhibit text?
011040' 037 00 0 00 013255'                     TMSGC   <CRAM (Correct):  >             ; no
011041' 200 00 0 00 011162'                     MOVE    SAVCR1                          ; get CRAM data written
011042' 260 17 0 00 011070'                     GO      MPNTC                           ; print it
011043' 603 01 0 00 000200                      TLNE    1,TXTINH                        ; inhibit text?
011044' 037 00 0 00 013261'                     TMSGC   <     (A):  >                   ; yes
011045' 607 01 0 00 000200                      TLNN    1,TXTINH                        ; inhibit text?
011046' 037 00 0 00 013264'                     TMSGC   <        (Actual):  >           ; no
011047' 200 00 0 00 011163'                     MOVE    SAVCR2                          ; get CRAM data
011050' 260 17 0 00 011070'                     GO      MPNTC                           ; print it
011051' 263 17 0 00 000000                      RTN



                                        ;#*********************************************************************
                                        ; MPNT2 - Print LAR data correct/actual (data in SAVCR1,SAVCR2)
                                        ;#*********************************************************************

011052' 200 01 0 00 011023*             MPNT2:  MOVE    1,SCOSW                         ; get switches
011053' 603 01 0 00 000200                      TLNE    1,TXTINH                        ; full printout?
011054' 037 00 0 00 013270'                     TMSGC   <LAR (C): >                     ; no
011055' 607 01 0 00 000200                      TLNN    1,TXTINH                        ; full printout?
011056' 037 00 0 00 013273'                     TMSGC   <LAR (Correct): >               ; yes
011057' 200 00 0 00 011162'                     MOVE    SAVCR1                          ; get address
011060' 037 04 0 00 000000                      PNT4                                    ; print it
011061' 603 01 0 00 000200                      TLNE    1,TXTINH                        ; full printout?
011062' 037 00 0 00 013277'                     TMSGC   <    (A): >                     ; no
011063' 607 01 0 00 000200                      TLNN    1,TXTINH                        ; full printout?
011064' 037 00 0 00 013302'                     TMSGC   <    (Actual):  >               ; yes
011065' 200 00 0 00 011163'                     MOVE    SAVCR2                          ; get address
011066' 037 04 0 00 000000                      PNT4                                    ; print it
011067' 263 17 0 00 000000                      RTN
```

```

                                        ;#*****************************************************************
                                        ; MPNTC - Print cram half word (30 bits only)
                                        ;#*****************************************************************

011070' 261 17 0 00 000001              MPNTC:  RPUT    (1,2)                   ; save AC's
011072' 200 01 0 00 000000                      MOVE    1,0                     ; get data into AC1
011073' 242 01 0 00 000006                      LSH     1,6                     ; position correctly
011074' 201 02 0 00 000012                      MOVEI   2,^D10                  ; set up to print 10 digits
011075' 400 00 0 00 000000              MPNTC0: SETZ                            ; clear AC1
011076' 246 00 0 00 000003                      LSHC    0,3                     ; shift digit into AC0
011077' 037 16 0 00 000003                      PNTOCS                          ; print it
011100' 367 02 0 00 011075'                     SOJG    2,MPNTC0                ; loop till done
011101' 262 17 0 00 000002                      RGET    (2,1)                   ; restore AC's
011103' 263 17 0 00 000000                      RTN                             ; return


                                        ;#*****************************************************************
                                        ; MPMBU1 - Print MBUS Error message
                                        ;#*****************************************************************

011104' 331 00 0 00 000000*             MPMBU1: SKIPL   ALFLS                   ; started up the port ok?
011105' 254 00 0 00 011117'                     JRST    MPMB1A                  ; yes - other message
011106' 200 00 0 00 011052*                     MOVE    SCOSW                   ; get switches
011107' 607 00 0 00 000200                      TLNN    TXTINH                  ; text inhibit switch set?
011110' 037 00 0 00 013306'                     TMSGC   <Couldn't start the port properly>
011111' 260 17 0 00 010760*                     GO      RDCSR                   ; get CSR contents
011112' 255 00 0 00 000000                      JFCL                            ; error
011113' 201 02 0 00 000023                      MOVEI   2,^D19                  ; character count so far
011114' 201 03 0 00 000023                      MOVEI   3,^D19                  ; continuation line column number
011115' 260 17 0 00 001112*                     GO      CSRPNT                  ; go print in English
011116' 263 17 0 00 000000                      RTN                             ; return

011117' 200 00 0 00 011106*             MPMB1A: MOVE    SCOSW                   ; get switches
011120' 603 00 0 00 000200                      TLNE    TXTINH                  ; text inhibit switch set?
                                                JRST    [TMSGC  <Set >          ; yes - just print what signal
                                                         PNTMSG @1              ;   was turned on
011121' 254 00 0 00 013317'                              JRST    MPMB1B]
011122' 037 00 0 00 013322'                     TMSGC   <Turned on MBUS driver with >
011123' 550 01 0 00 011165'                     HRRZ    1,SAVDAT                ; get ASCIZ string address
011124' 037 00 1 00 000001                      PNTMSG  @1                      ; print it
011125' 037 00 0 00 013330'                     TMSGC   <'MBUS Error' should not have occurred>
011126' 200 01 0 00 001670*             MPMB1B: MOVE    1,ALCSR                 ; get CSR data
011127' 201 02 0 00 000023                      MOVEI   2,^D19                  ; character count so far
011130' 201 03 0 00 000023                      MOVEI   3,^D19                  ; continuation line column number
011131' 260 17 0 00 011115*                     GO      CSRPNT                  ; go print in English
011132' 263 17 0 00 000000                      RTN                             ; return
```

```
                                        ;#*******************************************************************
                                        ; MPMBU2 - Print MBUS Error message
                                        ;#*******************************************************************

 011133' 331 00 0 00 011104*            MPMBU2: SKIPL   ALFLS                   ; started up the port ok?
 011134' 254 00 0 00 011146'                    JRST    MPMB2A                  ; yes - other message
 011135' 200 00 0 00 011117*                    MOVE    SCOSW                   ; get switches
 011136' 607 00 0 00 000200                     TLNN    TXTINH                  ; text inhibit switch set?
 011137' 037 00 0 00 013306'                    TMSGC   <Couldn't start the port properly>
 011140' 260 17 0 00 011111*                    GO      RDCSR                   ; get CSR contents
 011141' 255 00 0 00 000000                     JFCL                            ; error
 011142' 201 02 0 00 000023                     MOVEI   2,^D19                  ; character count so far
 011143' 201 03 0 00 000023                     MOVEI   3,^D19                  ; continuation line column number
 011144' 260 17 0 00 011131*                    GO      CSRPNT                  ; go print in English
 011145' 263 17 0 00 000000                     RTN                             ; return

 011146' 200 00 0 00 011135*            MPMB2A: MOVE    SCOSW                   ; get switches
 011147' 603 00 0 00 000200                     TLNE    TXTINH                  ; text inhibit switch set?
                                                JRST    [TMSGC  <Set >          ; yes - just print what signal
                                                         PNTMSG @1              ;   was turned on
 011150' 254 00 0 00 013340'                             JRST   MPMB2B]
 011151' 037 00 0 00 013322'                    TMSGC   <Turned on MBUS driver with >
 011152' 550 01 0 00 011165'                    HRRZ    1,SAVDAT                ; get ASCIZ string address
 011153' 037 00 1 00 000001                     PNTMSG  @1                      ; print it
 011154' 037 00 0 00 013343'                    TMSGC   <'MBUS Error' should have occurred>
 011155' 200 01 0 00 011126*            MPMB2B: MOVE    1,ALCSR                 ; get CSR data
 011156' 201 02 0 00 000023                     MOVEI   2,^D19                  ; character count so far
 011157' 201 03 0 00 000023                     MOVEI   3,^D19                  ; continuation line column number
 011160' 260 17 0 00 011144*                    GO      CSRPNT                  ; go print in English
 011161' 263 17 0 00 000000                     RTN                             ; return

                                        ;Miscellaneous variables

 011162' 000000  000000                 SAVCR1: 0
 011163' 000000  000000                 SAVCR2: 0
 011164' 000000  000000                 SAVCRA: 0
 011165' 000000  000000                 SAVDAT: 0
```

```

                                        ;#*********************************************************************
                                        ; End of MPROC Module Tests
                                        ;#*********************************************************************

                                                XLIST

NO ERRORS DETECTED

PROGRAM BREAK IS 013423
CPU TIME USED 08:08.087

229P CORE USED
```

```
DFPTA8  MACRO %53B(1242) 09:19  1-Jul-85 Page S-1
DFPTA8  MAC      18-Apr-85 10:59          SYMBOL TABLE                                                              SEQ 1543

AAPNT           000000  ext    GO      260740  000000            MA22LP          004137'          MBUS    010000  000000  spd
ADRPE   100000  000000  spd    INITPI          004726' ext       MA23            004325'          MLAST   400000  000000  spd
AEBUF           013106' ext    IPACLR          010741' ext       MA23L           004333'          MPER    200000  000000  spd
AEXEC           007162' ext    IPAER   000200  000000  spd       MA23LP          004330'          MPMB1A          011117'
ALCSR           011155' ext    IPASST          001057' ext       MA24            004537'          MPMB1B          011126'
ALFLS           011133' ext    LAST    010000  000000  spd       MA24L           004545'          MPMB2A          011146'
BBPNT           000000  ext    LAXER   000400  000000  spd       MA24LP          004542'          MPMB2B          011155'
BEXEC           010537' ext    LDCRAM          010755' ext       MA25            004731'          MPMBU1          011104'
BUFF            006664' ext    LDCSR           010757' ext       MA25L           004737'          MPMBU2          011133'
BUFGEN          006376' ext    LDRAR           010753' ext       MA25LP          004734'          MPNT1           011023'
C1              000054  spd    LGWC    000100  000000  spd       MA26            005107'          MPNT2           011052'
C10             000065  spd    M1              000031  spd       MA26L           005115'          MPNTC           011070'
C11             000066  spd    M10             000042  spd       MA26LP          005112'          MPNTCO          011075'
C12             000067  spd    M11             000043  spd       MA27            005320'          MPROC   200000  000000  spd
C13             000070  spd    M12             000044  spd       MA27L           005326'          MPRUN           000010  spd
C15             000072  spd    M13             000045  spd       MA27LP          005323'          MSG     100000  000000  spd
C16             000073  spd    M14             000046  spd       MA3             000303'          NDCB    000200  000000  spd
C17             000074  spd    M15             000047  spd       MA30            005600'          NDMP    000400  000000  spd
C21             000100  spd    M16             000050  spd       MA30L           005606'          NEXM    020000  000000  spd
C5              000060  spd    M17             000051  spd       MA30LP          005603'          NOTWCO  040000  000000  spd
C6              000061  spd    M18             000052  spd       MA30PN          005611'          ODELAY          010426' ext
C7              000062  spd    M19             000053  spd       MA31            006070'          OLDRAN          013421'
C8              000063  spd    M2              000032  spd       MA31L           006076'          OVN     000020  000000  spd
C9              000064  spd    M3              000033  spd       MA31L1          006120'          P               000017
CADDR           010703' ext    M4              000034  spd       MA31L2          006130'          PARFLG          010451' ext
CALL    200000  000000  spd    M41PNT          011000'           MA31LP          006073'          PAT             000014
CALPAR          010747' ext    M5              000035  spd       MA32            006406'          PNT3    037140  000000
CBUF            011645' ext    M6              000036  spd       MA32L           006414'          PNT4    037200  000000
CBUFSZ          000315  spd    M7              000037  spd       MA32LP          006411'          PNTDEC  037640  000000
CEBUF           013110' ext    M9              000041  spd       MA33            006670'          PNTHW   037540  000000
CHDATA          006611' ext    MA1             000051'           MA33L           006676'          PNTMSG  037000  000000
CHINIT          006663' ext    MA10            001377'           MA33LP          006673'          PNTOCS  037700  000003
CRAMPE  004000  000000  spd    MA11            001477'           MA34            007113'          PORTCI          010433' ext
CSRPNT          011160' ext    MA12            001561'           MA34PN          007115'          PORTNI          010431' ext
CWORDL          013113' ext    MA13            001707'           MA35            007170'          PUT     261740  000000
CWORDR          011004' ext    MA14            002044'           MA35P1          007174'          RANDBS          030022
DRCRAM          011001' ext    MA14L           002052'           MA35PN          007173'          RDCRAM          000455' ext
DWCRAM          010446' ext    MA14LP          002047'           MA36            010470'          RDCSR           011140' ext
E1              000001  spd    MA15            002320'           MA36PN          010472'          RDEBUF          007202' ext
E14             000016  spd    MA15L           002326'           MA37            010555'          RDLAR           004533' ext
E15             000017  spd    MA15LP          002323'           MA4             000370'          RTN     263740  000000
E16             000020  spd    MA16            002546'           MA40            010654'          SAVADD          013422'
E17             000021  spd    MA16PN          002551'           MA41            010774'          SAVCR1          011162'
E2              000002  spd    MA17            003372'           MA4PNT          000373'          SAVCR2          011163'
E23             000027  spd    MA17L           003400'           MA5             000474'          SAVCRA          011164'
E3              000003  spd    MA17LP          003375'           MA5PNT          000477'          SAVDA1          007237'
E4              000004  spd    MA2             000226'           MA6             000572'          SAVDAT          011165'
E6              000006  spd    MA20            003574'           MA6PNO          000606'          SCOPER  027000  000000
E7              000007  spd    MA20L           003602'           MA6PNT          000573'          SCOSW           011146' ext
E8              000010  spd    MA20LP          003577'           MA7             001100'          SDATA           004725' ext
E9              000011  spd    MA21            003743'           MA7PNO          001114'          SELLAR          040000  spd
ERFLG           000015         MA21L           003751'           MA7PNT          001101'          SETVEC          004727' ext
GENCCW          006666' ext    MA21LP          003746'           MB41            010776'          SEXEC           010625' ext
GENEPE          100000  spd    MA22            004134'           MBCN            000016           SHWC    000040  000000  spd
GET     262740  000000         MA22L           004142'           MBERR   002000  000000  spd      SINCYC          020000  spd
```

```
SLAST        011343' ext   TA15         002254'       TC5          000443'       TL23         004300'
SNEXT        001056' ext   TA16         002512'       TCALL        000003  spd   TL24         004471'
SSCALL       000003  spd   TA17         003340'       TCALLC       000004  spd   TL25         004664'
SSCHK        000004  spd   TA2          000167'       TCHECK       000005  spd   TL26         005054'
SSJRST       000005  spd   TA20         003550'       TDATAI       002313'       TL27         005235'
SSLAST       000000  spd   TA21         003722'       TDATAO       002315'       TL30         005530'
SSPNT        000000  ext   TA22         004113'       TEXEC        006624' ext   TL31         005763'
SSSTRT       000001  spd   TA23         004304'       TEXIT        000007  spd   TL32         006324'
T10M         001400'       TA24         004475'       TG1          000011'       TL33         006617'
T10TAB       001372'       TA25         004670'       TG10         001335'       TL34         007042'
T11M         001500'       TA26         005060'       TG11         001440'       TL35         007153'
T11TAB       001475'       TA27         005247'       TG12         001515'       TL36         010336'
T12M         001562'       TA3          000241'       TG13         001650'       TL37         010533'
T12TAB       001550'       TA30         005534'       TG14         001745'       TL40         010621'
T13M         001710'       TA31         005767'       TG15         002243'       TL41         010705'
T13TAB       001703'       TA32         006340'       TG16         002501'       TL6          000541'
T14M         002060'       TA33         006623'       TG17         003327'       TL7          001047'
T15M         002334'       TA34         007046'       TG2          000164'       TLAST        000000  spd
T16M         002560'       TA36         010342'       TG20         003537'       TLOAD        010617' ext
T17M         003404'       TA37         010537'       TG21         003711'       TRACE        010700' ext
T1PAT        000053'       TA4          000315'       TG22         004102'       TS14         001770'
T20M         003604'       TA40         010625'       TG23         004273'       TS14IN       002021'
T21M         003753'       TA41         010712'       TG24         004464'       TS14RW       002031'
T22M         004144'       TA5          000413'       TG25         004657'       TS15         002266'
T23M         004335'       TA6          000544'       TG26         005047'       TS16         002524'
T24M         004552'       TA6A         000562'       TG27         005230'       TS16AD       002531'
T25M         004742'       TA7          001052'       TG3          000237'       TS16AX       002545'
T26M         005121'       TA7A         001070'       TG30         005523'       TS17         003352'
T27M         005333'       TADDR        006622' ext   TG31         005756'       TS17PE       003367'
T27SR1       005305'       TB1          000027'       TG32         006317'       TS20         003562'
T27SR2       005307'       TB14         001757'       TG33         006612'       TS20PE       003571'
T30M         005640'       TB15         002255'       TG34         007035'       TS21         003734'
T30SR1       005562'       TB16         002513'       TG35         007146'       TS22         004125'
T30SRD       005560'       TB17         003341'       TG36         010334'       TS23         004316'
T31M         006140'       TB1L         000034'       TG37         010526'       TS24         004507'
T32M         006420'       TB2          000203'       TG4          000313'       TS24DI       004533'
T33M         006702'       TB20         003551'       TG40         010614'       TS24DO       004535'
T33SR1       006656'       TB21         003723'       TG41         010677'       TS25         004702'
T33SR2       006660'       TB22         004114'       TG5          000411'       TS25EX       004717'
T34M         007122'       TB23         004305'       TG6          000520'       TS25IA       004722'
T34MA        007130'       TB24         004476'       TG7          001026'       TS25IB       004725'
T35M         007243'       TB25         004671'       TI6A         000525'       TS26         005072'
T36MA        010503'       TB26         005061'       TI7A         001033'       TS27         005261'
T36MB        010514'       TB27         005250'       TJRST        000006  spd   TS30         005546'
T37M         010560'       TB2L         000210'       TL10         001342'       TS31         006001'
T40M         010656'       TB3          000245'       TL11         001445'       TS32         006352'
T41RAN       011007'       TB30         005535'       TL12         001522'       TS32RD       006373'
T6TAB        000625'       TB31         005770'       TL13         001655'       TS33         006635'
T7TAB        001133'       TB32         006341'       TL14         001752'       TS34         007057'
TA1          000014'       TB33         006624'       TL15         002250'       TS34IN       007067'
TA10         001345'       TB4          000321'       TL16         002506'       TS34L        007060'
TA11         001450'       TB41         010741'       TL17         003334'       TS34ST       007071'
TA12         001525'       TB5          000417'       TL20         003544'       TS36         010353'
TA13         001660'       TC3          000261'       TL21         003716'       TS36L        010354'
TA14         001756'       TC4          000342'       TL22         004107'       TS36LO       010402'
```

```
TS36L1          010420'       TX10             001371'         .LCCCE          000000  spd     .MCCEB           000037  spd
TS36L2          010442'       TX11             001474'         .LCCCP          000000  spd     .MCCER           000037  spd
TS36L3          010450'       TX12             001547'         .LCCEB          000000  spd     .MCCFZ           000037  spd
TS36LD          010361'       TX13             001702'         .LCCER          000000  spd     .MCCGC           000037  spd
TS36ST          010457'       TX14             001767'         .LCCFZ          000000  spd     .MCCIA           000037  spd
TS36W0          010453'       TX15             002265'         .LCCGC          000000  spd     .MCCLW           000037  spd
TS36W1          010455'       TX16             002523'         .LCCIA          000000  spd     .MCCMB           000037  spd
TS37            010550'       TX17             003351'         .LCCLW          000000  spd     .MCCMP           000037  spd
TS40            010636'       TX2              000225'         .LCCMB          000000  spd     .MCCPP           000037  spd
TS40A           010640'       TX20             003561'         .LCCMP          000000  spd     .MCCXA           000037  spd
TS40CH          010647'       TX21             003733'         .LCCPP          000000  spd     .MCENA           000001  spd
TS40IN          010644'       TX22             004124'         .LCCXA          000000  spd     .MCJP            000017  spd
TSLOD1          010432' ext   TX23             004315'         .LCENA          000000  spd     .MCJS            000017  spd
TSLOD2          010434' ext   TX24             004506'         .LCJP           000000  spd     .MCONT           000017  spd
TSTART          000001  spd   TX25             004701'         .LCJS           000000  spd     .MCRTN           000017  spd
TSTC3           000000  ext   TX26             005071'         .LCONT          000000  spd     .MCRY            000001  spd
TSTC4           000000  ext   TX27             005260'         .LCRTN          000000  spd     .MD              000007  spd
TSTE45          000000  ext   TX3              000302'         .LCRY           000000  spd     .MJ              007777  spd
TSTEBF          200000  spd   TX30             005545'         .LD             000000  spd     .MJMAP           000017  spd
TSTSUB          010705' ext   TX31             006000'         .LJ             010000  spd     .MJZ             000017  spd
TSTU1           000000' ent   TX32             006351'         .LJMAP          000000  spd     .MLDCT           000017  spd
TSTU10          001325' ent   TX33             006634'         .LJZ            000000  spd     .MLDLM           000037  spd
TSTU11          001431' ent   TX34             007056'         .LLDCT          000000  spd     .MLSAD           000037  spd
TSTU12          001507' ent   TX35             007167'         .LLDLM          000000  spd     .MMGC            001777  spd
TSTU13          001643' ent   TX36             010352'         .LLSAD          000000  spd     .MOENA           000001  spd
TSTU14          001733' ent   TX37             010547'         .LMGC           000001  spd     .MOR             000007  spd
TSTU15          002231' ent   TX4              000367'         .LOENA          002000  spd     .MPLUS           000007  spd
TSTU16          002467' ent   TX40             010635'         .LOR            000000  spd     .MRAM            000001  spd
TSTU17          003315' ent   TX41             010773'         .LPLUS          000000  spd     .MRDLM           000037  spd
TSTU2           000154' ent   TX5              000473'         .LRAM           000000  spd     .MRPCT           000017  spd
TSTU20          005525' ent   TX6              000571'         .LRDLM          000000  spd     .MSOA            000007  spd
TSTU21          003677' ent   TX7              001077'         .LRPCT          000000  spd     .MSOB            000007  spd
TSTU22          004070' ent   TXALL   060000   000000  spd     .LSOA           000000  spd     .MSAB            000007  spd
TSTU23          004261' ent   TXNOT   040000   000000  spd     .LSOB           000000  spd     .MSDO            000007  spd
TSTU24          004452' ent   TXTINH           000200  spd     .LSAB           000000  spd     .MSELC           000007  spd
TSTU25          004645' ent   UDEBUG           007074' ext     .LSDO           000000  spd     .MSELE           000007  spd
TSTU26          005035' ent   USER             030037          .LSELC          000000  spd     .MSELF           000007  spd
TSTU27          005216' ent   Z8               000000'         .LSELE          000000  spd     .MSELM           000007  spd
TSTU3           000230' ent   ZMPROC  000003   000000  spd     .LSELF          000000  spd     .MSELP           000007  spd
TSTU30          005512' ent   $ARG2            000001          .LSELM          000000  spd     .MSKCN           000037  spd
TSTU31          005747' ent   $B               000041          .LSELP          000000  spd     .MSKMB           000037  spd
TSTU32          006307' ent   $CHR    000057   455641          .LSKCN          000000  spd     .MXNOR           000007  spd
TSTU33          006603' ent   $GARG            000000          .LSKMB          000000  spd     .MXOR            000007  spd
TSTU34          007025' ent   %ADDR            000005  spd     .LXNOR          000000  spd     .RA     000010   000000  spd
TSTU35          007137' ent   %ML     000403   772000  spd     .LXOR           000000  spd     .RAND   040000   000000  spd
TSTU36          010326' ent   %MR     103010   210040  spd     .MA             000017  spd     .RB              400000  spd
TSTU37          010521' ent   .LA              000000  spd     .MAND           000007  spd     .RBAD            000001  spd
TSTU4           000305' ent   .LADDR  000100   000000  spd     .MB             000017  spd     .RCCAF           050000  spd
TSTU40          010607' ent   .LAND            000000  spd     .MBAD           000001  spd     .RCCBF           060000  spd
TSTU41          010671' ent   .LB              000000  spd     .MCCAF          000037  spd     .RCCCA           000000  spd
TSTU5           000404' ent   .LBAD            000000  spd     .MCCBF          000037  spd     .RCCCC           030000  spd
TSTU6           000510' ent   .LCCAF           000000  spd     .MCCCA          000037  spd     .RCCCE           160000  spd
TSTU7           001017' ent   .LCCBF           000000  spd     .MCCCC          000037  spd     .RCCCP           140000  spd
TTPNT           000000  ext   .LCCCA           000000  spd     .MCCCE          000037  spd     .RCCEB           040000  spd
TX1             000050'       .LCCCC           000000  spd     .MCCCP          000037  spd     .RCCER           100000  spd
```

```
.RCCFZ          020000  spd
.RCCGC          010000  spd
.RCCIA          110000  spd
.RCCLW          170000  spd
.RCCMB          120000  spd
.RCCMP          130000  spd
.RCCPP          150000  spd
.RCCXA          070000  spd
.RCENA  000400  000000  spd
.RCJP           000060  spd
.RCJS           000020  spd
.RCONT          000340  spd
.RCRTN          000240  spd
.RCRY           000400  spd
.RD     001000  000000  spd
.RJ             000000  spd
.RJMAP          000040  spd
.RJZ            000000  spd
.RLDCT          000300  spd
.RLDLM          230000  spd
.RLSAD          200000  spd
.RMGC           000000  spd
.ROENA          000000  spd
.ROR    030000  000000  spd
.RPLUS          000000  spd
.RRAM   000200  000000  spd
.RRDLM          220000  spd
.RRPCT          000220  spd
.RSOA   400000  000000  spd
.RSOB   300000  000000  spd
.RSAB   100000  000000  spd
.RSDO   700000  000000  spd
.RSELC          004000  spd
.RSELE          005000  spd
.RSELF          003000  spd
.RSELM          002000  spd
.RSELP          001000  spd
.RSKCN          240000  spd
.RSKMB          210000  spd
.RXNOR  070000  000000  spd
.RXOR   060000  000000  spd
```

```
AAPNT      24#   5896
ADRPE    4744
AEBUF      25#   6603
AEXEC      24#   1177   1295   1387   1531   5880
ALCSR      24#   1179   1297   1389   1533   7207   7238
ALFLS      24#   7186   7217
BBPNT      25#   5801   6727   6835
BEXEC      25#   5745   6600   6807
BUFF       22#   1722   4439   4442   4710   4711   4717   5298   5568   5571
BUFGEN     36#   1719   5295
C1       1619    5210   5492
C10      4885
C11      4346
C12      4641
C13      3151    3343   3535
C15      1619
C16      4346
C17      4641
C21      4885
C5       1619
C6       5210    5492
C7       1619    3151   3343   4641   5210   5492
C8       1619    4346   4885
C9       4641
CADDR      23#    663    664    912    913   6638   6642   6652   6663   6690   6696   7012
CALL      137     300    388    490    491    610    611    716    965   1209   1324   1420   1560   1731
         1732    2006   2007   2234   2235   2832   2833   3040   3041   3211   3212   3403   3404   3595
         3596    3822   3823   4027   4028   4219   4220   4451   4452   4726   4727   5000   5001   5307
         5308    5579   5580   5800   5801   5896   5897   6726   6727   6835   6957   7088   7091
CALPAR     23#   7031   7057
CBUF       22#   1720   4440   4715   5296   5569
CBUFSZ   1720    4440   4715   5296   5569
CEBUF      25#   6605
CHDATA     32#   1738
CHINIT     32#   1721   4441   4716   5297   5570
CRAMPE    697     946   1182   1300   7042   7068
CSRPNT     22#    727    976   7195   7210   7226   7241
CWORDL     23#    668    917   6640   6644   6656   6658   6661   6662   6699   7020   7028   7054   7095
         7113
CWORDR     23#    669    918   7021   7029   7055   7097
DRCRAM     23#   6639   6643   6691   7094
DWCRAM     23#    672    921   6659   6664   6700
E1       2752    2974   3731
E14      3941
E15       648     898   1149   1268   1361   1506   3941
E16       898    2752   2974
E17      3731
E2       3731
E23      2167    4146
E3       1912
E4       2974
E6       2752    2974   3941
E7       1912    3941
```

```
E8          2752
E9          2974
ERFLG        114       119       123       276       282       286       341       372       376       438       450       454       458       471
             475       552       567       571       575       591       595       685       691       692       693       694       696       698
             702       718       934       940       941       942       943       945       947       951       967      1165      1181      1183
            1187      1283      1299      1301      1305      1375      1391      1395      1519      1535      1539      1637      1648      1655
            1930      1941      1948      2185      2196      2203      2770      2781      2788      2992      3003      3010      3169      3180
            3187      3361      3372      3379      3553      3564      3571      3749      3760      3767      3959      3970      3977      4164
            4175      4182      4364      4381      4388      4658      4669      4676      4900      4911      4918      5226      5245      5252
            5507      5518      5525      5738      5748      5755      5873      5881      6593      6606      6614      6800      6810      6817
            6912      6922      6929      7018      7041      7043      7047      7067      7069      7073
GENCCW        32#     1725      4445      4720      5301      5573
GENEPE      2825      3033
INITPI        32#     4020
IPACLR        31#       97       258       342       431       545       658       688       907       937      1166      1284      1376      1520
            1644      1707      1713      1937      2192      2777      2999      3176      3368      3560      3756      3966      4171      4365
            4377      4665      4907      5230      5241      5514      5739      5874      6594      6801      6913      7025      7051
IPAER       4752
IPASST        31#      690       939
LAST         137       300       388       491       611       716       965      1209      1324      1420      1560      1732      2007      2235
            2833      3041      3212      3404      3596      3823      4028      4220      4452      4727      5001      5308      5580      5801
            5897      6727      6835      6957      7088      7091
LAXER       4750
LDCRAM        30#      115       277       278       351       355       446       467      7033      7037      7059      7063
LDCSR         30#      433       547       560       563       584       587      1712      2826      3034      4373      5237      5903      5905
            6675      6679      7039      7065
LDRAR         30#      101       263       350       353       368       444       465       558       582      1710      2000      3816      4371
            5235      5901      6673      7027      7035      7053      7061
LGWC        4754
M1            75       237
M10           75       237       329
M11         5722      5858      6582      6787      6899
M12          648       898      6999
M13         5722      5858      6582      6787
M14         5722      5858      6582      6787
M15         5722      5858      6582      6787
M16           75       237       329       420       535
M17           75       237       329
M18           75       237       329
M19           75       237       329
M2          5722      5858      6582      6787
M3          1149      1268      1361      1506
M4            75       237       329       420       535
M41PNT      7088      7091      7093#
M5          6899
M6          1619      1912      2167      2752      2974      3151      3343      3535      3731      3941      4146      4346      4641      4885
            5210      5492
M7          1619      1912      2167      2752      2974      3151      3343      3535      3731      3941      4146      4346      4641      4885
            5210      5492
M9           420       535
MA1          120       136#
MA10        1184      1209#
MA11        1302      1324#
```

```
MA12        1392      1420#
```

```
MA12       1392      1420#
MA13       1536      1560#
MA14       1652      1730#
MA14L      1735      1738#
MA14LP     1731      1734#
MA15       1945      2005#
MA15L      2010      2013#
MA15LP     2006      2009#
MA16       2200      2233#
MA16PN     2234      2237#
MA17       2785      2831#
MA17L      2836      2839#
MA17LP     2832      2835#
MA2         283       299#
MA20       3007      3039#
MA20L      3044      3047#
MA20LP     3040      3043#
MA21       3184      3210#
MA21L      3215      3218#
MA21LP     3211      3214#
MA22       3376      3402#
MA22L      3407      3410#
MA22LP     3403      3406#
MA23       3568      3594#
MA23L      3599      3602#
MA23LP     3595      3598#
MA24       3764      3821#
MA24L      3826      3829#
MA24LP     3822      3825#
MA25       3974      4026#
MA25L      4031      4034#
MA25LP     4027      4030#
MA26       4179      4218#
MA26L      4223      4226#
MA26LP     4219      4222#
MA27       4385      4450#
MA27L      4455      4458#
MA27LP     4451      4454#
MA3         373       387#
MA30       4673      4725#
MA30L      4730      4733#
MA30LP     4726      4729#
MA30PN     4738#
MA31       4915      4999#
MA31L      5004      5007#
MA31L1     5008      5009      5010      5011      5012      5013      5014      5015      5026#
MA31L2     5016      5017      5018      5019      5020      5021      5022      5023      5035#
MA31LP     5000      5003#
MA32       5249      5306#
MA32L      5311      5314#
MA32LP     5307      5310#
MA33       5522      5578#
MA33L      5583      5586#
```

```
MA33LP     5579      5582#
```

```
MA33LP   5579      5582#
MA34     5752      5800#
MA34PN   5800      5803#
MA35     5885      5895#
MA35P1   5900#     5910
MA35PN   5897      5899#
MA36     6611      6726#
MA36PN   6726      6729#
MA37     6814      6833#
MA4       451       472       489#
MA40     6926      6956#
MA41     7044      7087#
MA4PNT    490       495#
MA5       568       592       609#
MA5PNT    610       615#
MA6       699       716#
MA6PNO    719       730#
MA6PNT    716       718#
MA7       948       965#
MA7PNO    968       979#
MA7PNT    965       967#
MB41     7070      7090#
MBCN     5789      5791      6631      6633      6683      6685
MBERR    1180      1298      1390      1534
MBUS      418       533      1147      1266      1359      1504      5720      6897
MLAST      75       237       329       420       535       648       898      1149      1268      1361      1506      1619      1912      2167
         2752      2974      3151      3343      3535      3731      3941      4146      4346      4641      4885      5210      5492      5722
         5858      6582      6787      6899      6999
MPER     4742
MPMB1A   7187      7198#
MPMB1B   7202      7207#
MPMB2A   7218      7229#
MPMB2B   7233      7238#
MPMBU1   1209      1324      7186#
MPMBU2   1420      1560      7217#
MPNT1     137       300       388      7122#
MPNT2     491       611      7151#
MPNTC    7096      7098      7137      7143      7170#
MPNTCO   7174#     7177
MPROC      73       235       327       418       533       646       896      1147      1266      1359      1504      1617      1910      2165
         2750      2972      3149      3341      3533      3729      3939      4144      4344      4639      4883      5208      5490      5720
         5856      6580      6785      6897      6997
MPRUN     680       929      1711      2825      3033      4372      5236      5902      6674      7038      7064
MSG       136       299       387       489       609      1730      2005      2233      2831      3039      3210      3402      3594      3821
         4026      4218      4450      4725      4999      5306      5578      5895      6833      6834      6956
NDCB     1266      1504
NDMP       73       235       327       418       533       646       896      1147      1266      1359      1504      1617      1910      2165
         2750      2972      3149      3341      3533      3729      3939      4144      4344      4639      4883      5208      5490      5720
         5856      6580      6785      6897      6997
NEXM     4748
NOTWCO   4746
ODELAY     22#     6677
OLDRAN   7008#     7008      7105      7109
```

```
OVN     4758
P       2219    4010    5779    5782    5795    6718    6950
PARFLG    23#    670     671     673     919     920     922    6646    6647    6704
PAT     1716    5292
PORTCI    24#   5791    6633    6685
PORTNI    24#   5789    6631    6683
RANDBS  7007    7103    7107
RDCRAM    30#    116     279     369     561     585
RDCSR     30#    695     723     944     972    7040    7066    7191    7222
RDEBUF    30#   5906
RDLAR     30#    447     468     564     588    1996    3813
SAVADD  6944#   6944    6945    6948
SAVCR1   107     118     269     281     367     371     441     449     462     470     495     498     555     566
         579     590     615     618    7136    7156    7246#
SAVCR2   117     280     370     448     469     565     589    7142    7162    7247#
SAVCRA   100     262     366    7124    7127    7248#
SAVDA1  5907    5917    5918    5927    5933    5937#
SAVDAT  1175    1293    1385    1529    2220#   2220    2222    2228    2237    2241    5774    5777    5784    5804#
        5804    6715    6716    6720    6733#   6733    7204    7235    7249#
SCOSW     36#    720     730     969     979    5912    5922    6729    7122    7151    7188    7198    7219    7229
SDATA     31#    681     930    4016    4019
SELLAR   432     546     562     586    2825    3033
SETVEC    32#   4021
SEXEC     26#   6919
SHWC    4756
SINCYC  7038    7064
SLAST     31#    733     737     740     744     982     986     989     993
SNEXT     31#    689     938
SSCALL  5763    5764    6622    6623    6937    6938
SSCHK   5764    5765    6623    6624    6939    6940
SSJRST  5769    5770    6626    6627    6941
SSLAST  5770    5771    6627    6628    6829    6830    6942    6943
SSPNT     26#   6957
SSSTRT  5765    5766    5767    5768    5769    6624    6625    6626    6825    6826    6827    6828    6829    6938
        6939    6940    6941
T10M    1148    1159    1213#
T10TAB  1174    1192    1201#
T11M    1267    1277    1328#
T11TAB  1292    1310    1319#
T12M    1360    1369    1424#
T12TAB  1384    1400    1409#
T13M    1505    1513    1564#
T13TAB  1528    1544    1553#
T14M    1618    1631    1749#
T15M    1911    1924    2026#
T16M    2166    2179    2248#
T17M    2751    2764    2848#
T1PAT    106     141#    268
T20M    2973    2986    3054#
T21M    3150    3163    3225#
T22M    3342    3355    3417#
T23M    3534    3547    3609#
T24M    3730    3743    3839#
```

```
T25M      3940   3953   4043#
T26M      4145   4158   4235#
T27M      4345   4358   4472#
T27SR1    4405   4421   4428   4436#  5535   5542   5556
T27SR2    4414   4438#  5549
T30M      4640   4652   4766#
T30SR1    4708#  4714
T30SRD    4692   4706#
T31M      4884   4894   5048#
T32M      5209   5220   5324#
T33M      5491   5501   5595#
T33SR1    5565#
T33SR2    5567#
T34M      5721   5732   5785   5786   5793   5811#
T34MA     5787   5788   5818#
T35M      5857   5867   5941#
T36MA     6635   6741#
T36MB     6687   6753#
T37M      6786   6794   6841#
T40M      6898   6906   6961#
T41RAN    7019   7103#
T6TAB      662    751#
T7TAB      911   1000#
TA1         90#   109
TA10      1171#  1185   1193
TA11      1289#  1303   1311
TA12      1381#  1393   1401
TA13      1525#  1537   1545
TA14      1644#  1655
TA15      1937#  1948
TA16      2192#  2203
TA17      2777#  2788
TA2        251#   271
TA20      2999#  3010
TA21      3176#  3187
TA22      3368#  3379
TA23      3560#  3571
TA24      3756#  3767
TA25      3966#  3977
TA26      4171#  4182
TA27      4377#
TA3        341#   374
TA30      4665#  4676
TA31      4907#  4918
TA32      5241#  5252
TA33      5514#  5525
TA34      5745#  5747   5755
TA36      6600#  6602   6614
TA37      6807#  6809   6817
TA4        431#
TA40      6919#  6921   6929
TA41      7025#  7045
TA5        545#
```

```
TA6        685#   700    708
```

1540

```
                                                                                                      SEQ 1553
TA6         685#    700     708
TA6A        694     699#
TA7         934#    949     957
TA7A        943     948#
TADDR        26#   1640    1933    2188    2217    2773    2995    3172    3364    3556    3752    3962    4167    4368
           4661    4903    5229    5232    5510
TB1         106#    128
TB14       1645#   1647
TB15       1938#   1940
TB16       2193#   2195
TB17       2778#   2780
TB1L        114#    121
TB2         268#    291
TB20       3000#   3002
TB21       3177#   3179
TB22       3369#   3371
TB23       3561#   3563
TB24       3757#   3759
TB25       3967#   3969
TB26       4172#   4174
TB27       4378#   4380    4388
TB2L        276#    284
TB3         348#    358
TB30       4666#   4668
TB31       4908#   4910
TB32       5242#   5244
TB33       5515#   5517
TB4         438#    452     481
TB41       7051#   7071
TB5         552#    569     601
TC3         363#    379
TC4         458#    473
TC5         575#    593
TCALL      1665    1666    1672    1673    1679    1680    1687    1688    1694    1695    1701    1702    1984    1985
           1991    1992    2811    2812    3027    3028    3784    3785    3791    3792    3799    3800    3806    3807
           3993    3994    4001    4002    4405    4406    4414    4415    4421    4422    4428    4429    4692    4693
           5262    5263    5269    5270    5277    5278    5284    5285    5535    5536    5542    5543    5549    5550
           5556    5557
TCALLC     2211    2212    3997    3998
TCHECK     1667    1668    1674    1675    1681    1682    1689    1690    1696    1697    1703    1704    1959    1960
           1965    1966    1971    1972    1978    1979    1985    1986    1992    1993    2213    2214    2799    2800
           2805    2806    2812    2813    2819    2820    3021    3022    3028    3029    3198    3199    3204    3205
           3390    3391    3396    3397    3582    3583    3588    3589    3778    3779    3785    3786    3792    3793
           3800    3801    3807    3808    3988    3989    3995    3996    4003    4004    4193    4194    4199    4200
           4205    4206    4212    4213    4399    4400    4407    4408    4416    4417    4423    4424    4430    4431
           4687    4688    4694    4695    4700    4701    4929    4930    4936    4937    4939    4940    4942    4943
           4945    4946    4948    4949    4952    4953    4955    4956    4958    4959    4966    4967    4969    4970
           4972    4973    4975    4976    4978    4979    4981    4982    4984    4985    4987    4988    4993    4994
           5264    5265    5271    5272    5279    5280    5286    5287    5537    5538    5544    5545    5551    5552
           5558    5559
TDATAI     1984    1996#
TDATAO     1991    1999#
TEXEC        26#   1645    1938    2193    2778    3000    3177    3369    3561    3757    3967    4172    4378    4666
```

```
            4908     5242     5515
TEXIT       1668     1669     1675     1676     1682     1683     1690     1691     1697     1698     1704     1705     1960     1961
            1966     1967     1972     1973     1979     1980     1986     1987     1993     1994     2214     2215     2800     2801
            2806     2807     2813     2814     3022     3023     3199     3200     3205     3206     3391     3392     3397     3398
            3583     3584     3589     3590     3779     3780     3786     3787     3793     3794     3801     3802     3808     3809
            3989     3990     3996     3997     4004     4005     4194     4195     4200     4201     4206     4207     4213     4214
            4400     4401     4408     4409     4417     4418     4424     4425     4431     4432     4688     4689     4695     4696
            4930     4931     4937     4938     4940     4941     4943     4944     4946     4947     4949     4950     4953     4954
            4956     4957     4959     4960     4967     4968     4970     4971     4973     4974     4976     4977     4979     4980
            4982     4983     4985     4986     4988     4989     4994     4995     5265     5266     5272     5273     5280     5281
            5287     5288     5538     5539     5545     5546     5552     5553     5559     5560
TG1           72       84#
TG10        1146     1157#
TG11        1265     1275#
TG12        1358     1367#
TG13        1503     1511#
TG14        1616     1629#
TG15        1909     1922#
TG16        2164     2177#
TG17        2749     2762#
TG2          234      245#
TG20        2971     2984#
TG21        3148     3161#
TG22        3340     3353#
TG23        3532     3545#
TG24        3728     3741#
TG25        3938     3951#
TG26        4143     4156#
TG27        4343     4356#
TG3          326      336#
TG30        4638     4650#
TG31        4882     4892#
TG32        5207     5218#
TG33        5489     5499#
TG34        5719     5730#
TG35        5855     5865#
TG36        6579     6588#
TG37        6784     6792#
TG4          417      426#
TG40        6896     6904#
TG41        6996     7005#
TG5          532      540#
TG6          645      656#
TG7          895      905#
TI6A         664#     675
TI7A         913#     924
TJRST       2215     2216
TL10        1165#
TL11        1283#
TL12        1375#
TL13        1519#
TL14        1637#    1653
TL15        1930#    1946
```

```
TL16      2185#  2201
TL17      2770#  2786
TL20      2992#  3008
TL21      3169#  3185
TL22      3361#  3377
TL23      3553#  3569
TL24      3749#  3765
TL25      3959#  3975
TL26      4164#  4180
TL27      4364#  4386
TL30      4658#  4674
TL31      4900#  4916
TL32      5226#  5250
TL33      5507#  5523
TL34      5738#  5753
TL35      5873#  5886
TL36      6593#  6612
TL37      6800#  6815
TL40      6912#  6927
TL41      7014#  7079
TL6        667    679#
TL7        916    928#
TLAST     1705   1706    1994    1995    2821    2822    3029    3030    3206    3207    3398    3399    3590    3591
          3809   3810    4005    4006    4214    4215    4432    4433    4701    4702    4995    4996    5288    5289
          5560   5561
TLOAD       22#  1160    1278    1370    1514    1632    1925    2180    2765    2987    3164    3356    3548    3744
          3954   4159    4359    4653    4895    5221    5502    5733    5794    5868    6636    6688    6795    6907
TRACE       22#    85     246     337     427     541     657     906    1158    1276    1368    1512    1630    1923
          2178   2763    2985    3162    3354    3546    3742    3952    4157    4357    4651    4893    5219    5500
          5731   5866    6589    6793    6905    7006
TS14      1638   1665#
TS14IN    1665   1707#
TS14RW    1672   1679    1687    1694    1701    1716#
TS15      1931   1958#
TS16      2186   2211#   2215
TS16AD    2211   2217#
TS16AX    2221   2229#
TS17      2771   2798#
TS17PE    2811   2825#
TS20      2993   3020#
TS20PE    3027   3033#
TS21      3170   3197#
TS22      3362   3389#
TS23      3554   3581#
TS24      3750   3777#
TS24DI    3784   3799    3813#
TS24DO    3791   3806    3816#
TS25      3960   3987#
TS25EX    3997   4009#
TS25IA    3993   4015#
TS25IB    4001   4019#
TS26      4165   4192#
TS27      4367   4398#
```

```
TS30       4659       4686#
TS31       4901       4928#
TS32       5227       5231       5262#
TS32RD     5262       5269       5277       5284       5292#
TS33       5508       5535#
TS34       5741       5763#
TS34IN     5763       5774#
TS34L      5764#      5769
TS34ST     5764       5777#
TS36       6596       6622#
TS36L      6623#      6626       6719       6721
TS36L0     6652#      6665
TS36L1     6654       6671#
TS36L2     6696#      6701
TS36L3     6698       6703#
TS36LD     6622       6631#
TS36ST     6623       6714#
TS36W0     6641       6655       6709#
TS36W1     6645       6660       6710#
TS37       6803       6825#
TS40       6915       6937#
TS40A      6939#      6941       6951
TS40CH     6939       6948#
TS40IN     6937       6944#
TSLOD1       23#      5790       6632       6684
TSLOD2       23#      5792       6634       6686
TSTART     1666       1667       1673       1674       1680       1681       1688       1689       1695       1696       1702       1703       1958       1959
           1964       1965       1970       1971       1977       1978       1983       1984       1990       1991       2212       2213       2798       2799
           2804       2805       2810       2811       2818       2819       3020       3021       3026       3027       3197       3198       3203       3204
           3389       3390       3395       3396       3581       3582       3587       3588       3777       3778       3783       3784       3790       3791
           3798       3799       3805       3806       3987       3988       3994       3995       4002       4003       4192       4193       4198       4199
           4204       4205       4211       4212       4398       4399       4406       4407       4415       4416       4422       4423       4429       4430
           4686       4687       4693       4694       4699       4700       4928       4929       4935       4936       4938       4939       4941       4942
           4944       4945       4947       4948       4951       4952       4954       4955       4957       4958       4965       4966       4968       4969
           4971       4972       4974       4975       4977       4978       4980       4981       4983       4984       4986       4987       4992       4993
           5263       5264       5270       5271       5278       5279       5285       5286       5536       5537       5543       5544       5550       5551
           5557       5558
TSTC3        18#      1620       4347       4642       5211       5493
TSTC4        18#      1621       4348       4643       5212       5494
TSTE45       18#       650        899       7000
TSTEBF     5904
TSTSUB       22#       113        275        343        363        442        443        463        464        556        557        580        581        687
            936       1171       1172       1289       1290       1381       1382       1525       1526       1639       1734       1932       2009       2187
           2772       2835       2994       3043       3171       3214       3363       3406       3555       3598       3751       3825       3961       4030
           4166       4222       4366       4454       4660       4729       4902       5003       5027       5036       5228       5310       5509       5582
           5740       5875       6595       6802       6914       7014
TSTU1        10         72#
TSTU10       10       1146#
TSTU11       11       1150       1265#
TSTU12       11       1151       1269       1358#
TSTU13       11       1152       1270       1362       1503#
TSTU14       11       1616#
TSTU15       11       1909#
```

```
TSTU16      11     1913   2164#
TSTU17      11     1914   2168    2749#
TSTU2       10       76    234#
TSTU20      11     1915   2169    2753    2971#
TSTU21      12     1916   2170    2754    2975    3148#
TSTU22      12     1917   2171    2755    2976    3152    3340#
TSTU23      12     2172   2756    2977    3153    3344    3532#
TSTU24      12     2757   2978    3154    3345    3536    3728#
TSTU25      12     2979   3155    3346    3537    3732    3938#
TSTU26      12     3156   3347    3538    3733    3942    4143#
TSTU27      12     3348   3539    3734    3943    4147    4343#
TSTU3       10       77    238     326#
TSTU30      12     1622   3540    3735    3944    4148    4349    4638#
TSTU31      13     3736   3945    4149    4882#
TSTU32      13     1623   3946    4150    4350    4644    4886    5207#
TSTU33      13     1624   4151    4351    4645    4887    5213    5489#
TSTU34      13     5719#
TSTU35      13     5723   5855#
TSTU36      13     5724   5859    6579#
TSTU37      13     5725   5860    6583    6784#
TSTU4       10       78    239     330     417#
TSTU40      13     6896#
TSTU41      14      651    900    6996#
TSTU5       10       79    240     331     421     532#
TSTU6       10      645#
TSTU7       10      649    895#
TTPNT       26#    1732   2007    2235    2833    3041    3212    3404    3596    3823    4028    4220    4452    4727
          5001     5308   5580
TX1         92       95    122     123     132#
TX10      1186     1187   1197#
TX11      1304     1305   1315#
TX12      1394     1395   1405#
TX13      1538     1539   1549#
TX14      1646     1654   1659#
TX15      1939     1947   1952#
TX16      2194     2202   2207#
TX17      2779     2787   2792#
TX2        253      256    285     286     295#
TX20      3001     3009   3014#
TX21      3178     3186   3191#
TX22      3370     3378   3383#
TX23      3562     3570   3575#
TX24      3758     3766   3771#
TX25      3968     3976   3981#
TX26      4173     4181   4186#
TX27      4379     4387   4392#
TX3        375      376    383#
TX30      4667     4675   4680#
TX31      4909     4917   4922#
TX32      5243     5251   5256#
TX33      5516     5524   5529#
TX34      5746     5754   5759#
TX35      5887     5891#
```

```
TX36        6601    6613    6618#
TX37        6808    6816    6821#
TX4          453     454     474     475     485#
TX40        6920    6928    6933#
TX41        7046    7047    7072    7073    7083#
TX5          570     571     594     595     605#
TX6          701     702     712#
TX7          950     951     961#
TXALL        137     300     388     491     611     716     965    1209    1324    1420    1560    1732    2007    2235
            2833    3041    3212    3404    3596    3823    4028    4220    4452    4727    5001    5308    5580    5801
            5896    5897    6726    6727    6835    6957    7088    7091
TXNOT        136     299     387     489     490     609     610    1730    1731    2005    2006    2233    2234    2831
            2832    3039    3040    3210    3211    3402    3403    3594    3595    3821    3822    4026    4027    4218
            4219    4450    4451    4725    4726    4999    5000    5306    5307    5578    5579    5800    5895    6833
            6834    6956    7087    7090
TXTINH       721     731     970     980    5913    5915    5923    5925    5929    5931    6730    6735    7132    7134
            7138    7140    7152    7154    7158    7160    7189    7199    7220    7230
UDEBUG        22#     93     254    5780
USER        4009
Z8            43#     84     245     336     426     540     656     905    1157    1275    1367    1511    1629    1922
            2177    2762    2984    3161    3353    3545    3741    3951    4156    4356    4650    4892    5218    5499
            5730    5865    6588    6792    6904    7005
ZMPROC        73     235     327     418     533     646     896    1147    1266    1359    1504    1617    1910    2165
            2750    2972    3149    3341    3533    3729    3939    4144    4344    4639    4883    5208    5490    5720
            5856    6580    6785    6897    6997
$ARG2       1213#   1213    1215#   1215    1217#   1217    1220#   1220    1222#   1222    1224#   1224    1227#   1227
            1229#   1229    1231#   1231    1234#   1234    1236#   1236    1238#   1238    1328#   1328    1330#   1330
            1332#   1332    1424#   1424    1426#   1426    1428#   1428    1431#   1431    1433#   1433    1435#   1435
            1438#   1438    1440#   1440    1442#   1442    1445#   1445    1447#   1447    1449#   1449    1452#   1452
            1454#   1454    1456#   1456    1459#   1459    1461#   1461    1463#   1463    1466#   1466    1468#   1468
            1470#   1470    1473#   1473    1475#   1475    1477#   1477    1564#   1564    1566#   1566    1568#   1568
            1571#   1571    1573#   1573    1575#   1575    1578#   1578    1580#   1580    1582#   1582    1749#   1749
            1751#   1751    1753#   1753    1758#   1758    1760#   1760    1762#   1762    1764#   1764    1769#   1769
            1771#   1771    1773#   1773    1775#   1775    1780#   1780    1782#   1782    1784#   1784    1786#   1786
            1788#   1788    1790#   1790    1792#   1792    1797#   1797    1799#   1799    1801#   1801    1803#   1803
            1805#   1805    1807#   1807    1809#   1809    1814#   1814    1816#   1816    1818#   1818    1820#   1820
            1822#   1822    1824#   1824    1826#   1826    1831#   1831    1833#   1833    1838#   1838    1840#   1840
            1842#   1842    1844#   1844    1846#   1846    1848#   1848    1850#   1850    1852#   1852    1854#   1854
            1859#   1859    1861#   1861    1863#   1863    1865#   1865    1867#   1867    1869#   1869    1871#   1871
            1873#   1873    1875#   1875    2026#   2026    2028#   2028    2030#   2030    2035#   2035    2037#   2037
            2039#   2039    2044#   2044    2046#   2046    2048#   2048    2050#   2050    2052#   2052    2054#   2054
            2059#   2059    2061#   2061    2063#   2063    2065#   2065    2067#   2067    2069#   2069    2071#   2071
            2077#   2077    2079#   2079    2081#   2081    2083#   2083    2085#   2085    2087#   2087    2089#   2089
            2091#   2091    2096#   2096    2098#   2098    2100#   2100    2102#   2102    2104#   2104    2106#   2106
            2108#   2108    2110#   2110    2112#   2112    2117#   2117    2119#   2119    2121#   2121    2123#   2123
            2125#   2125    2127#   2127    2129#   2129    2131#   2131    2133#   2133    2248#   2248    2250#   2250
            2252#   2252    2254#   2254    2256#   2256    2258#   2258    2260#   2260    2262#   2262    2264#   2264
            2266#   2266    2268#   2268    2270#   2270    2272#   2272    2274#   2274    2276#   2276    2278#   2278
            2280#   2280    2282#   2282    2284#   2284    2286#   2286    2288#   2288    2290#   2290    2292#   2292
            2294#   2294    2296#   2296    2298#   2298    2300#   2300    2302#   2302    2304#   2304    2306#   2306
            2308#   2308    2310#   2310    2312#   2312    2314#   2314    2316#   2316    2318#   2318    2320#   2320
            2322#   2322    2327#   2327    2329#   2329    2331#   2331    2336#   2336    2338#   2338    2340#   2340
            2342#   2342    2347#   2347    2349#   2349    2351#   2351    2356#   2356    2358#   2358    2360#   2360
```

```
EQ 1546  2365#  2365   2367#  2367   2369#  2369   2374#  2374   2376#  2376   2378#  2378   2383#  2383   SEQ 1559
         2385#  2385   2387#  2387   2392#  2392   2394#  2394   2396#  2396   2401#  2401   2403#  2403
         2405#  2405   2410#  2410   2412#  2412   2414#  2414   2419#  2419   2421#  2421   2423#  2423
         2428#  2428   2430#  2430   2432#  2432   2437#  2437   2439#  2439   2441#  2441   2446#  2446
         2448#  2448   2450#  2450   2455#  2455   2457#  2457   2459#  2459   2464#  2464   2466#  2466
         2468#  2468   2473#  2473   2475#  2475   2477#  2477   2482#  2482   2484#  2484   2486#  2486
         2491#  2491   2493#  2493   2495#  2495   2500#  2500   2502#  2502   2504#  2504   2509#  2509
         2511#  2511   2513#  2513   2518#  2518   2520#  2520   2522#  2522   2527#  2527   2529#  2529
         2531#  2531   2536#  2536   2538#  2538   2540#  2540   2545#  2545   2547#  2547   2549#  2549
         2554#  2554   2556#  2556   2558#  2558   2563#  2563   2565#  2565   2567#  2567   2572#  2572
         2574#  2574   2576#  2576   2581#  2581   2583#  2583   2585#  2585   2590#  2590   2592#  2592
         2594#  2594   2599#  2599   2601#  2601   2603#  2603   2608#  2608   2610#  2610   2612#  2612
         2617#  2617   2619#  2619   2621#  2621   2626#  2626   2628#  2628   2630#  2630   2635#  2635
         2637#  2637   2639#  2639   2644#  2644   2646#  2646   2648#  2648   2653#  2653   2655#  2655
         2657#  2657   2659#  2659   2664#  2664   2666#  2666   2671#  2671   2673#  2673   2675#  2675
         2680#  2680   2682#  2682   2684#  2684   2686#  2686   2688#  2688   2690#  2690   2692#  2692
         2694#  2694   2696#  2696   2701#  2701   2703#  2703   2705#  2705   2707#  2707   2709#  2709
         2711#  2711   2713#  2713   2715#  2715   2717#  2717   2848#  2848   2850#  2850   2852#  2852
         2857#  2857   2859#  2859   2861#  2861   2863#  2863   2865#  2865   2867#  2867   2872#  2872
         2874#  2874   2876#  2876   2878#  2878   2880#  2880   2882#  2882   2887#  2887   2889#  2889
         2891#  2891   2893#  2893   2895#  2895   2897#  2897   2899#  2899   2904#  2904   2906#  2906
         2908#  2908   2910#  2910   2912#  2912   2914#  2914   2916#  2916   2918#  2918   2920#  2920
         2925#  2925   2927#  2927   2929#  2929   2931#  2931   2933#  2933   2935#  2935   2937#  2937
         2939#  2939   2941#  2941   3054#  3054   3056#  3056   3058#  3058   3063#  3063   3065#  3065
         3067#  3067   3069#  3069   3071#  3071   3073#  3073   3075#  3075   3077#  3077   3082#  3082
         3084#  3084   3086#  3086   3088#  3088   3090#  3090   3092#  3092   3094#  3094   3096#  3096
         3098#  3098   3103#  3103   3105#  3105   3107#  3107   3109#  3109   3111#  3111   3113#  3113
         3115#  3115   3117#  3117   3119#  3119   3225#  3225   3227#  3227   3229#  3229   3231#  3231
         3233#  3233   3235#  3235   3237#  3237   3239#  3239   3244#  3244   3246#  3246   3248#  3248
         3250#  3250   3252#  3252   3254#  3254   3259#  3259   3261#  3261   3263#  3263   3265#  3265
         3267#  3267   3269#  3269   3274#  3274   3276#  3276   3278#  3278   3280#  3280   3282#  3282
         3284#  3284   3286#  3286   3288#  3288   3290#  3290   3295#  3295   3297#  3297   3299#  3299
         3301#  3301   3303#  3303   3305#  3305   3307#  3307   3309#  3309   3311#  3311   3417#  3417
         3419#  3419   3421#  3421   3423#  3423   3425#  3425   3427#  3427   3429#  3429   3431#  3431
         3436#  3436   3438#  3438   3440#  3440   3442#  3442   3444#  3444   3446#  3446   3451#  3451
         3453#  3453   3455#  3455   3457#  3457   3459#  3459   3461#  3461   3466#  3466   3468#  3468
         3470#  3470   3472#  3472   3474#  3474   3476#  3476   3478#  3478   3480#  3480   3482#  3482
         3487#  3487   3489#  3489   3491#  3491   3493#  3493   3495#  3495   3497#  3497   3499#  3499
         3501#  3501   3503#  3503   3609#  3609   3611#  3611   3613#  3613   3615#  3615   3617#  3617
         3619#  3619   3621#  3621   3623#  3623   3628#  3628   3630#  3630   3632#  3632   3634#  3634
         3636#  3636   3638#  3638   3643#  3643   3645#  3645   3647#  3647   3649#  3649   3651#  3651
         3653#  3653   3658#  3658   3660#  3660   3662#  3662   3664#  3664   3666#  3666   3668#  3668
         3670#  3670   3672#  3672   3674#  3674   3679#  3679   3681#  3681   3683#  3683   3685#  3685
         3687#  3687   3689#  3689   3691#  3691   3693#  3693   3695#  3695   3839#  3839   3841#  3841
         3843#  3843   3848#  3848   3850#  3850   3852#  3852   3857#  3857   3859#  3859   3861#  3861
         3863#  3863   3865#  3865   3870#  3870   3872#  3872   3874#  3874   3876#  3876   3878#  3878
         3880#  3880   3882#  3882   3884#  3884   3886#  3886   3891#  3891   3893#  3893   3895#  3895
         3897#  3897   3899#  3899   3901#  3901   3903#  3903   3905#  3905   3907#  3907   4043#  4043
         4045#  4045   4047#  4047   4052#  4052   4054#  4054   4056#  4056   4058#  4058   4063#  4063
         4065#  4065   4067#  4067   4069#  4069   4074#  4074   4076#  4076   4078#  4078   4080#  4080
         4082#  4082   4084#  4084   4086#  4086   4088#  4088   4090#  4090   4095#  4095   4097#  4097
         4099#  4099   4101#  4101   4103#  4103   4105#  4105   4107#  4107   4109#  4109   4111#  4111
         4235#  4235   4237#  4237   4239#  4239   4244#  4244   4246#  4246   4248#  4248   4253#  4253
```

EQ 1547

SEQ 1560

```
4255#  4255   4257#  4257   4262#  4262   4264#  4264   4266#  4266   4271#  4271   4273#  4273
4275#  4275   4277#  4277   4279#  4279   4281#  4281   4283#  4283   4285#  4285   4287#  4287
4292#  4292   4294#  4294   4296#  4296   4298#  4298   4300#  4300   4302#  4302   4304#  4304
4306#  4306   4308#  4308   4472#  4472   4474#  4474   4476#  4476   4482#  4482   4484#  4484
4486#  4486   4488#  4488   4494#  4494   4496#  4496   4498#  4498   4500#  4500   4505#  4505
4507#  4507   4509#  4509   4511#  4511   4513#  4513   4518#  4518   4520#  4520   4522#  4522
4524#  4524   4526#  4526   4531#  4531   4533#  4533   4535#  4535   4540#  4540   4542#  4542
4544#  4544   4546#  4546   4548#  4548   4550#  4550   4552#  4552   4554#  4554   4556#  4556
4558#  4558   4560#  4560   4562#  4562   4564#  4564   4569#  4569   4571#  4571   4573#  4573
4575#  4575   4577#  4577   4579#  4579   4581#  4581   4583#  4583   4585#  4585   4590#  4590
4592#  4592   4594#  4594   4596#  4596   4598#  4598   4600#  4600   4602#  4602   4604#  4604
4606#  4606   4766#  4766   4768#  4768   4770#  4770   4776#  4776   4778#  4778   4780#  4780
4782#  4782   4784#  4784   4786#  4786   4788#  4788   4790#  4790   4792#  4792   4794#  4794
4799#  4799   4801#  4801   4803#  4803   4805#  4805   4810#  4810   4812#  4812   4814#  4814
4816#  4816   4818#  4818   4820#  4820   4822#  4822   4824#  4824   4826#  4826   4831#  4831
4833#  4833   4835#  4835   4837#  4837   4839#  4839   4841#  4841   4843#  4843   4845#  4845
4847#  4847   5048#  5048   5050#  5050   5052#  5052   5058#  5058   5060#  5060   5062#  5062
5064#  5064   5066#  5066   5068#  5068   5070#  5070   5072#  5072   5079#  5079   5081#  5081
5083#  5083   5085#  5085   5087#  5087   5089#  5089   5091#  5091   5093#  5093   5098#  5098
5100#  5100   5102#  5102   5104#  5104   5110#  5110   5112#  5112   5114#  5114   5116#  5116
5118#  5118   5124#  5124   5126#  5126   5128#  5128   5130#  5130   5132#  5132   5137#  5137
5139#  5139   5141#  5141   5143#  5143   5145#  5145   5147#  5147   5149#  5149   5151#  5151
5153#  5153   5158#  5158   5160#  5160   5162#  5162   5164#  5164   5166#  5166   5168#  5168
5170#  5170   5172#  5172   5174#  5174   5324#  5324   5326#  5326   5328#  5328   5330#  5330
5332#  5332   5337#  5337   5339#  5339   5341#  5341   5343#  5343   5345#  5345   5347#  5347
5349#  5349   5351#  5351   5353#  5353   5355#  5355   5360#  5360   5362#  5362   5364#  5364
5366#  5366   5368#  5368   5370#  5370   5372#  5372   5374#  5374   5379#  5379   5381#  5381
5383#  5383   5385#  5385   5387#  5387   5389#  5389   5391#  5391   5393#  5393   5398#  5398
5400#  5400   5402#  5402   5404#  5404   5406#  5406   5408#  5408   5413#  5413   5415#  5415
5420#  5420   5422#  5422   5424#  5424   5426#  5426   5428#  5428   5430#  5430   5432#  5432
5434#  5434   5436#  5436   5441#  5441   5443#  5443   5445#  5445   5447#  5447   5449#  5449
5451#  5451   5453#  5453   5455#  5455   5457#  5457   5595#  5595   5597#  5597   5599#  5599
5601#  5601   5606#  5606   5608#  5608   5610#  5610   5612#  5612   5614#  5614   5616#  5616
5621#  5621   5623#  5623   5625#  5625   5627#  5627   5629#  5629   5631#  5631   5636#  5636
5638#  5638   5640#  5640   5642#  5642   5644#  5644   5646#  5646   5648#  5648   5653#  5653
5655#  5655   5657#  5657   5659#  5659   5661#  5661   5663#  5663   5665#  5665   5667#  5667
5669#  5669   5674#  5674   5676#  5676   5678#  5678   5680#  5680   5682#  5682   5684#  5684
5686#  5686   5688#  5688   5690#  5690   5811#  5811   5813#  5813   5815#  5815   5818#  5818
5820#  5820   5822#  5822   5941#  5941   5946#  5946   5948#  5948   5950#  5950   5952#  5952
5954#  5954   5956#  5956   5958#  5958   5960#  5960   5962#  5962   5964#  5964   5966#  5966
5968#  5968   5970#  5970   5972#  5972   5974#  5974   5976#  5976   5978#  5978   5980#  5980
5982#  5982   5984#  5984   5986#  5986   5988#  5988   5990#  5990   5992#  5992   5994#  5994
5996#  5996   5998#  5998   6000#  6000   6002#  6002   6004#  6004   6006#  6006   6008#  6008
6010#  6010   6012#  6012   6014#  6014   6016#  6016   6018#  6018   6020#  6020   6022#  6022
6024#  6024   6026#  6026   6028#  6028   6030#  6030   6032#  6032   6034#  6034   6036#  6036
6038#  6038   6040#  6040   6042#  6042   6044#  6044   6046#  6046   6048#  6048   6050#  6050
6052#  6052   6054#  6054   6056#  6056   6058#  6058   6060#  6060   6062#  6062   6064#  6064
6066#  6066   6068#  6068   6070#  6070   6072#  6072   6074#  6074   6076#  6076   6078#  6078
6080#  6080   6082#  6082   6084#  6084   6086#  6086   6088#  6088   6090#  6090   6092#  6092
6094#  6094   6096#  6096   6098#  6098   6100#  6100   6102#  6102   6104#  6104   6106#  6106
6108#  6108   6110#  6110   6112#  6112   6114#  6114   6116#  6116   6118#  6118   6120#  6120
6122#  6122   6124#  6124   6126#  6126   6128#  6128   6130#  6130   6132#  6132   6134#  6134
6136#  6136   6138#  6138   6140#  6140   6142#  6142   6144#  6144   6146#  6146   6148#  6148
```

```
548       6150#  6150   6152#  6152   6154#  6154   6156#  6156   6158#  6158   6160#  6160   6162#  6162    SEQ 1561
          6164#  6164   6166#  6166   6168#  6168   6173#  6173   6181#  6181   6183#  6183   6185#  6185
          6187#  6187   6189#  6189   6194#  6194   6196#  6196   6198#  6198   6200#  6200   6202#  6202
          6204#  6204   6206#  6206   6208#  6208   6210#  6210   6212#  6212   6214#  6214   6216#  6216
          6218#  6218   6220#  6220   6222#  6222   6224#  6224   6226#  6226   6228#  6228   6230#  6230
          6232#  6232   6234#  6234   6236#  6236   6238#  6238   6240#  6240   6242#  6242   6244#  6244
          6246#  6246   6248#  6248   6250#  6250   6252#  6252   6254#  6254   6256#  6256   6258#  6258
          6263#  6263   6265#  6265   6267#  6267   6269#  6269   6272#  6272   6274#  6274   6276#  6276
          6278#  6278   6281#  6281   6283#  6283   6285#  6285   6287#  6287   6290#  6290   6292#  6292
          6294#  6294   6296#  6296   6299#  6299   6301#  6301   6303#  6303   6305#  6305   6308#  6308
          6310#  6310   6312#  6312   6314#  6314   6317#  6317   6319#  6319   6321#  6321   6323#  6323
          6326#  6326   6328#  6328   6330#  6330   6332#  6332   6335#  6335   6337#  6337   6339#  6339
          6341#  6341   6344#  6344   6346#  6346   6348#  6348   6350#  6350   6353#  6353   6355#  6355
          6357#  6357   6359#  6359   6362#  6362   6364#  6364   6366#  6366   6368#  6368   6371#  6371
          6373#  6373   6375#  6375   6377#  6377   6380#  6380   6382#  6382   6384#  6384   6386#  6386
          6389#  6389   6391#  6391   6393#  6393   6395#  6395   6398#  6398   6400#  6400   6402#  6402
          6404#  6404   6407#  6407   6409#  6409   6411#  6411   6413#  6413   6416#  6416   6418#  6418
          6420#  6420   6422#  6422   6425#  6425   6427#  6427   6429#  6429   6431#  6431   6434#  6434
          6436#  6436   6438#  6438   6440#  6440   6443#  6443   6445#  6445   6447#  6447   6449#  6449
          6452#  6452   6454#  6454   6456#  6456   6458#  6458   6461#  6461   6463#  6463   6465#  6465
          6467#  6467   6470#  6470   6472#  6472   6474#  6474   6476#  6476   6479#  6479   6481#  6481
          6483#  6483   6485#  6485   6488#  6488   6490#  6490   6492#  6492   6494#  6494   6497#  6497
          6499#  6499   6501#  6501   6503#  6503   6506#  6506   6508#  6508   6510#  6510   6512#  6512
          6515#  6515   6517#  6517   6519#  6519   6521#  6521   6524#  6524   6526#  6526   6528#  6528
          6530#  6530   6533#  6533   6535#  6535   6537#  6537   6539#  6539   6542#  6542   6544#  6544
          6546#  6546   6548#  6548   6551#  6551   6741#  6741   6743#  6743   6745#  6745   6747#  6747
          6753#  6753   6755#  6755   6841#  6841   6843#  6843   6845#  6845   6847#  6847   6852#  6852
          6854#  6854   6859#  6859   6861#  6861   6863#  6863   6865#  6865   6867#  6867   6961#  6961
          6963#  6963   6965#  6965   6967#  6967   6969#  6969
  $B      1213#  1213   1215#  1215   1217#  1217   1220#  1220   1222#  1222   1224#  1224   1227#  1227
          1229#  1229   1231#  1231   1234#  1234   1236#  1236   1238#  1238   1328#  1328   1330#  1330
          1332#  1332   1424#  1424   1426#  1426   1428#  1428   1431#  1431   1433#  1433   1435#  1435
          1438#  1438   1440#  1440   1442#  1442   1445#  1445   1447#  1447   1449#  1449   1452#  1452
          1454#  1454   1456#  1456   1459#  1459   1461#  1461   1463#  1463   1466#  1466   1468#  1468
          1470#  1470   1473#  1473   1475#  1475   1477#  1477   1564#  1564   1566#  1566   1568#  1568
          1571#  1571   1573#  1573   1575#  1575   1578#  1578   1580#  1580   1582#  1582   1749#  1749
          1751#  1751   1753#  1753   1758#  1758   1760#  1760   1762#  1762   1764#  1764   1769#  1769
          1771#  1771   1773#  1773   1775#  1775   1780#  1780   1782#  1782   1784#  1784   1786#  1786
          1788#  1788   1790#  1790   1792#  1792   1797#  1797   1799#  1799   1801#  1801   1803#  1803
          1805#  1805   1807#  1807   1809#  1809   1814#  1814   1816#  1816   1818#  1818   1820#  1820
          1822#  1822   1824#  1824   1826#  1826   1831#  1831   1833#  1833   1838#  1838   1840#  1840
          1842#  1842   1844#  1844   1846#  1846   1848#  1848   1850#  1850   1852#  1852   1854#  1854
          1859#  1859   1861#  1861   1863#  1863   1865#  1865   1867#  1867   1869#  1869   1871#  1871
          1873#  1873   1875#  1875   2026#  2026   2028#  2028   2030#  2030   2035#  2035   2037#  2037
          2039#  2039   2044#  2044   2046#  2046   2048#  2048   2050#  2050   2052#  2052   2054#  2054
          2059#  2059   2061#  2061   2063#  2063   2065#  2065   2067#  2067   2069#  2069   2071#  2071
          2077#  2077   2079#  2079   2081#  2081   2083#  2083   2085#  2085   2087#  2087   2089#  2089
          2091#  2091   2096#  2096   2098#  2098   2100#  2100   2102#  2102   2104#  2104   2106#  2106
          2108#  2108   2110#  2110   2112#  2112   2117#  2117   2119#  2119   2121#  2121   2123#  2123
          2125#  2125   2127#  2127   2129#  2129   2131#  2131   2133#  2133   2248#  2248   2250#  2250
          2252#  2252   2254#  2254   2256#  2256   2258#  2258   2260#  2260   2262#  2262   2264#  2264
          2266#  2266   2268#  2268   2270#  2270   2272#  2272   2274#  2274   2276#  2276   2278#  2278
          2280#  2280   2282#  2282   2284#  2284   2286#  2286   2288#  2288   2290#  2290   2292#  2292
```

```
549       2294#  2294   2296#  2296   2298#  2298   2300#  2300   2302#  2302   2304#  2304   2306#  2306    SEQ 1562
```

```
2294#  2294   2296#  2296   2298#  2298   2300#  2300   2302#  2302   2304#  2304   2306#  2306
2308#  2308   2310#  2310   2312#  2312   2314#  2314   2316#  2316   2318#  2318   2320#  2320
2322#  2322   2327#  2327   2329#  2329   2331#  2331   2336#  2336   2338#  2338   2340#  2340
2342#  2342   2347#  2347   2349#  2349   2351#  2351   2356#  2356   2358#  2358   2360#  2360
2365#  2365   2367#  2367   2369#  2369   2374#  2374   2376#  2376   2378#  2378   2383#  2383
2385#  2385   2387#  2387   2392#  2392   2394#  2394   2396#  2396   2401#  2401   2403#  2403
2405#  2405   2410#  2410   2412#  2412   2414#  2414   2419#  2419   2421#  2421   2423#  2423
2428#  2428   2430#  2430   2432#  2432   2437#  2437   2439#  2439   2441#  2441   2446#  2446
2448#  2448   2450#  2450   2455#  2455   2457#  2457   2459#  2459   2464#  2464   2466#  2466
2468#  2468   2473#  2473   2475#  2475   2477#  2477   2482#  2482   2484#  2484   2486#  2486
2491#  2491   2493#  2493   2495#  2495   2500#  2500   2502#  2502   2504#  2504   2509#  2509
2511#  2511   2513#  2513   2518#  2518   2520#  2520   2522#  2522   2527#  2527   2529#  2529
2531#  2531   2536#  2536   2538#  2538   2540#  2540   2545#  2545   2547#  2547   2549#  2549
2554#  2554   2556#  2556   2558#  2558   2563#  2563   2565#  2565   2567#  2567   2572#  2572
2574#  2574   2576#  2576   2581#  2581   2583#  2583   2585#  2585   2590#  2590   2592#  2592
2594#  2594   2599#  2599   2601#  2601   2603#  2603   2608#  2608   2610#  2610   2612#  2612
2617#  2617   2619#  2619   2621#  2621   2626#  2626   2628#  2628   2630#  2630   2635#  2635
2637#  2637   2639#  2639   2644#  2644   2646#  2646   2648#  2648   2653#  2653   2655#  2655
2657#  2657   2659#  2659   2664#  2664   2666#  2666   2671#  2671   2673#  2673   2675#  2675
2680#  2680   2682#  2682   2684#  2684   2686#  2686   2688#  2688   2690#  2690   2692#  2692
2694#  2694   2696#  2696   2701#  2701   2703#  2703   2705#  2705   2707#  2707   2709#  2709
2711#  2711   2713#  2713   2715#  2715   2717#  2717   2848#  2848   2850#  2850   2852#  2852
2857#  2857   2859#  2859   2861#  2861   2863#  2863   2865#  2865   2867#  2867   2872#  2872
2874#  2874   2876#  2876   2878#  2878   2880#  2880   2882#  2882   2887#  2887   2889#  2889
2891#  2891   2893#  2893   2895#  2895   2897#  2897   2899#  2899   2904#  2904   2906#  2906
2908#  2908   2910#  2910   2912#  2912   2914#  2914   2916#  2916   2918#  2918   2920#  2920
2925#  2925   2927#  2927   2929#  2929   2931#  2931   2933#  2933   2935#  2935   2937#  2937
2939#  2939   2941#  2941   3054#  3054   3056#  3056   3058#  3058   3063#  3063   3065#  3065
3067#  3067   3069#  3069   3071#  3071   3073#  3073   3075#  3075   3077#  3077   3082#  3082
3084#  3084   3086#  3086   3088#  3088   3090#  3090   3092#  3092   3094#  3094   3096#  3096
3098#  3098   3103#  3103   3105#  3105   3107#  3107   3109#  3109   3111#  3111   3113#  3113
3115#  3115   3117#  3117   3119#  3119   3225#  3225   3227#  3227   3229#  3229   3231#  3231
3233#  3233   3235#  3235   3237#  3237   3239#  3239   3244#  3244   3246#  3246   3248#  3248
3250#  3250   3252#  3252   3254#  3254   3259#  3259   3261#  3261   3263#  3263   3265#  3265
3267#  3267   3269#  3269   3274#  3274   3276#  3276   3278#  3278   3280#  3280   3282#  3282
3284#  3284   3286#  3286   3288#  3288   3290#  3290   3295#  3295   3297#  3297   3299#  3299
3301#  3301   3303#  3303   3305#  3305   3307#  3307   3309#  3309   3311#  3311   3417#  3417
3419#  3419   3421#  3421   3423#  3423   3425#  3425   3427#  3427   3429#  3429   3431#  3431
3436#  3436   3438#  3438   3440#  3440   3442#  3442   3444#  3444   3446#  3446   3451#  3451
3453#  3453   3455#  3455   3457#  3457   3459#  3459   3461#  3461   3466#  3466   3468#  3468
3470#  3470   3472#  3472   3474#  3474   3476#  3476   3478#  3478   3480#  3480   3482#  3482
3487#  3487   3489#  3489   3491#  3491   3493#  3493   3495#  3495   3497#  3497   3499#  3499
3501#  3501   3503#  3503   3609#  3609   3611#  3611   3613#  3613   3615#  3615   3617#  3617
3619#  3619   3621#  3621   3623#  3623   3628#  3628   3630#  3630   3632#  3632   3634#  3634
3636#  3636   3638#  3638   3643#  3643   3645#  3645   3647#  3647   3649#  3649   3651#  3651
3653#  3653   3658#  3658   3660#  3660   3662#  3662   3664#  3664   3666#  3666   3668#  3668
3670#  3670   3672#  3672   3674#  3674   3679#  3679   3681#  3681   3683#  3683   3685#  3685
3687#  3687   3689#  3689   3691#  3691   3693#  3693   3695#  3695   3839#  3839   3841#  3841
3843#  3843   3848#  3848   3850#  3850   3852#  3852   3857#  3857   3859#  3859   3861#  3861
3863#  3863   3865#  3865   3870#  3870   3872#  3872   3874#  3874   3876#  3876   3878#  3878
3880#  3880   3882#  3882   3884#  3884   3886#  3886   3891#  3891   3893#  3893   3895#  3895
3897#  3897   3899#  3899   3901#  3901   3903#  3903   3905#  3905   3907#  3907   4043#  4043
4045#  4045   4047#  4047   4052#  4052   4054#  4054   4056#  4056   4058#  4058   4063#  4063
```

```
4065#  4065   4067#  4067   4069#  4069   4074#  4074   4076#  4076   4078#  4078   4080#  4080
```

550

```
4065#  4065   4067#  4067   4069#  4069   4074#  4074   4076#  4076   4078#  4078   4080#  4080
4082#  4082   4084#  4084   4086#  4086   4088#  4088   4090#  4090   4095#  4095   4097#  4097
4099#  4099   4101#  4101   4103#  4103   4105#  4105   4107#  4107   4109#  4109   4111#  4111
4235#  4235   4237#  4237   4239#  4239   4244#  4244   4246#  4246   4248#  4248   4253#  4253
4255#  4255   4257#  4257   4262#  4262   4264#  4264   4266#  4266   4271#  4271   4273#  4273
4275#  4275   4277#  4277   4279#  4279   4281#  4281   4283#  4283   4285#  4285   4287#  4287
4292#  4292   4294#  4294   4296#  4296   4298#  4298   4300#  4300   4302#  4302   4304#  5304
4306#  4306   4308#  4308   4472#  4472   4474#  4474   4476#  4476   4482#  4482   4484#  4484
4486#  4486   4488#  4488   4494#  4494   4496#  4496   4498#  4498   4500#  4500   4505#  4505
4507#  4507   4509#  4509   4511#  4511   4513#  4513   4518#  4518   4520#  4520   4522#  4522
4524#  4524   4526#  4526   4531#  4531   4533#  4533   4535#  4535   4540#  4540   4542#  4542
4544#  4544   4546#  4546   4548#  4548   4550#  4550   4552#  4552   4554#  4554   4556#  4556
4558#  4558   4560#  4560   4562#  4562   4564#  4564   4569#  4569   4571#  4571   4573#  4573
4575#  4575   4577#  4577   4579#  4579   4581#  4581   4583#  4583   4585#  4585   4590#  4590
4592#  4592   4594#  4594   4596#  4596   4598#  4598   4600#  4600   4602#  4602   4604#  4604
4606#  4606   4766#  4766   4768#  4768   4770#  4770   4776#  4776   4778#  4778   4780#  4780
4782#  4782   4784#  4784   4786#  4786   4788#  4788   4790#  4790   4792#  4792   4794#  4794
4799#  4799   4801#  4801   4803#  4803   4805#  4805   4810#  4810   4812#  4812   4814#  4814
4816#  4816   4818#  4818   4820#  4820   4822#  4822   4824#  4824   4826#  4826   4831#  4831
4833#  4833   4835#  4835   4837#  4837   4839#  4839   4841#  4841   4843#  4843   4845#  4845
4847#  4847   5048#  5048   5050#  5050   5052#  5052   5058#  5058   5060#  5060   5062#  5062
5064#  5064   5066#  5066   5068#  5068   5070#  5070   5072#  5072   5079#  5079   5081#  5081
5083#  5083   5085#  5085   5087#  5087   5089#  5089   5091#  5091   5093#  5093   5098#  5098
5100#  5100   5102#  5102   5104#  5104   5110#  5110   5112#  5112   5114#  5114   5116#  5116
5118#  5118   5124#  5124   5126#  5126   5128#  5128   5130#  5130   5132#  5132   5137#  5137
5139#  5139   5141#  5141   5143#  5143   5145#  5145   5147#  5147   5149#  5149   5151#  5151
5153#  5153   5158#  5158   5160#  5160   5162#  5162   5164#  5164   5166#  5166   5168#  5168
5170#  5170   5172#  5172   5174#  5174   5324#  5324   5326#  5326   5328#  5328   5330#  5330
5332#  5332   5337#  5337   5339#  5339   5341#  5341   5343#  5343   5345#  5345   5347#  5347
5349#  5349   5351#  5351   5353#  5353   5355#  5355   5360#  5360   5362#  5362   5364#  5364
5366#  5366   5368#  5368   5370#  5370   5372#  5372   5374#  5374   5379#  5379   5381#  5381
5383#  5383   5385#  5385   5387#  5387   5389#  5389   5391#  5391   5393#  5393   5398#  5398
5400#  5400   5402#  5402   5404#  5404   5406#  5406   5408#  5408   5413#  5413   5415#  5415
5420#  5420   5422#  5422   5424#  5424   5426#  5426   5428#  5428   5430#  5430   5432#  5432
5434#  5434   5436#  5436   5441#  5441   5443#  5443   5445#  5445   5447#  5447   5449#  5449
5451#  5451   5453#  5453   5455#  5455   5457#  5457   5595#  5595   5597#  5597   5599#  5599
5601#  5601   5606#  5606   5608#  5608   5610#  5610   5612#  5612   5614#  5614   5616#  5616
5621#  5621   5623#  5623   5625#  5625   5627#  5627   5629#  5629   5631#  5631   5636#  5636
5638#  5638   5640#  5640   5642#  5642   5644#  5644   5646#  5646   5648#  5648   5653#  5653
5655#  5655   5657#  5657   5659#  5659   5661#  5661   5663#  5663   5665#  5665   5667#  5667
5669#  5669   5674#  5674   5676#  5676   5678#  5678   5680#  5680   5682#  5682   5684#  5684
5686#  5686   5688#  5688   5690#  5690   5811#  5811   5813#  5813   5815#  5815   5818#  5818
5820#  5820   5822#  5822   5941#  5941   5946#  5946   5948#  5948   5950#  5950   5952#  5952
5954#  5954   5956#  5956   5958#  5958   5960#  5960   5962#  5962   5964#  5964   5966#  5966
5968#  5968   5970#  5970   5972#  5972   5974#  5974   5976#  5976   5978#  5978   5980#  5980
5982#  5982   5984#  5984   5986#  5986   5988#  5988   5990#  5990   5992#  5992   5994#  5994
5996#  5996   5998#  5998   6000#  6000   6002#  6002   6004#  6004   6006#  6006   6008#  6008
6010#  6010   6012#  6012   6014#  6014   6016#  6016   6018#  6018   6020#  6020   6022#  6022
6024#  6024   6026#  6026   6028#  6028   6030#  6030   6032#  6032   6034#  6034   6036#  6036
6038#  6038   6040#  6040   6042#  6042   6044#  6044   6046#  6046   6048#  6048   6050#  6050
6052#  6052   6054#  6054   6056#  6056   6058#  6058   6060#  6060   6062#  6062   6064#  6064
6066#  6066   6068#  6068   6070#  6070   6072#  6072   6074#  6074   6076#  6076   6078#  6078
6080#  6080   6082#  6082   6084#  6084   6086#  6086   6088#  6088   6090#  6090   6092#  6092
```

%MR

```
6094#  6094   6096#  6096   6098#  6098   6100#  6100   6102#  6102   6104#  6104   6106#  6106
```

```
1551
              6094#  6094    6096#  6096    6098#  6098    6100#  6100    6102#  6102    6104#  6104    6106#  6106      SEQ 1564
              6108#  6108    6110#  6110    6112#  6112    6114#  6114    6116#  6116    6118#  6118    6120#  6120
              6122#  6122    6124#  6124    6126#  6126    6128#  6128    6130#  6130    6132#  6132    6134#  6134
              6136#  6136    6138#  6138    6140#  6140    6142#  6142    6144#  6144    6146#  6146    6148#  6148
              6150#  6150    6152#  6152    6154#  6154    6156#  6156    6158#  6158    6160#  6160    6162#  6162
              6164#  6164    6166#  6166    6168#  6168    6173#  6173    6181#  6181    6183#  6183    6185#  6185
              6187#  6187    6189#  6189    6194#  6194    6196#  6196    6198#  6198    6200#  6200    6202#  6202
              6204#  6204    6206#  6206    6208#  6208    6210#  6210    6212#  6212    6214#  6214    6216#  6216
              6218#  6218    6220#  6220    6222#  6222    6224#  6224    6226#  6226    6228#  6228    6230#  6230
              6232#  6232    6234#  6234    6236#  6236    6238#  6238    6240#  6240    6242#  6242    6244#  6244
              6246#  6246    6248#  6248    6250#  6250    6252#  6252    6254#  6254    6256#  6256    6258#  6258
              6263#  6263    6265#  6265    6267#  6267    6269#  6269    6272#  6272    6274#  6274    6276#  6276
              6278#  6278    6281#  6281    6283#  6283    6285#  6285    6287#  6287    6290#  6290    6292#  6292
              6294#  6294    6296#  6296    6299#  6299    6301#  6301    6303#  6303    6305#  6305    6308#  6308
              6310#  6310    6312#  6312    6314#  6314    6317#  6317    6319#  6319    6321#  6321    6323#  6323
              6326#  6326    6328#  6328    6330#  6330    6332#  6332    6335#  6335    6337#  6337    6339#  6339
              6341#  6341    6344#  6344    6346#  6346    6348#  6348    6350#  6350    6353#  6353    6355#  6355
              6357#  6357    6359#  6359    6362#  6362    6364#  6364    6366#  6366    6368#  6368    6371#  6371
              6373#  6373    6375#  6375    6377#  6377    6380#  6380    6382#  6382    6384#  6384    6386#  6386
              6389#  6389    6391#  6391    6393#  6393    6395#  6395    6398#  6398    6400#  6400    6402#  6402
              6404#  6404    6407#  6407    6409#  6409    6411#  6411    6413#  6413    6416#  6416    6418#  6418
              6420#  6420    6422#  6422    6425#  6425    6427#  6427    6429#  6429    6431#  6431    6434#  6434
              6436#  6436    6438#  6438    6440#  6440    6443#  6443    6445#  6445    6447#  6447    6449#  6449
              6452#  6452    6454#  6454    6456#  6456    6458#  6458    6461#  6461    6463#  6463    6465#  6465
              6467#  6467    6470#  6470    6472#  6472    6474#  6474    6476#  6476    6479#  6479    6481#  6481
              6483#  6483    6485#  6485    6488#  6488    6490#  6490    6492#  6492    6494#  6494    6497#  6497
              6499#  6499    6501#  6501    6503#  6503    6506#  6506    6508#  6508    6510#  6510    6512#  6512
              6515#  6515    6517#  6517    6519#  6519    6521#  6521    6524#  6524    6526#  6526    6528#  6528
              6530#  6530    6533#  6533    6535#  6535    6537#  6537    6539#  6539    6542#  6542    6544#  6544
              6546#  6546    6548#  6548    6551#  6551    6741#  6741    6743#  6743    6745#  6745    6747#  6747
              6753#  6753    6755#  6755    6841#  6841    6843#  6843    6845#  6845    6847#  6847    6852#  6852
              6854#  6854    6859#  6859    6861#  6861    6863#  6863    6865#  6865    6867#  6867    6961#  6961
              6963#  6963    6965#  6965    6967#  6967    6969#  6969
   $CHR       1213#  1213    1215#  1215    1217#  1217    1220#  1220    1222#  1222    1224#  1224    1227#  1227
              1229#  1229    1231#  1231    1234#  1234    1236#  1236    1238#  1238    1328#  1328    1330#  1330
              1332#  1332    1424#  1424    1426#  1426    1428#  1428    1431#  1431    1433#  1433    1435#  1435
              1438#  1438    1440#  1440    1442#  1442    1445#  1445    1447#  1447    1449#  1449    1452#  1452
              1454#  1454    1456#  1456    1459#  1459    1461#  1461    1463#  1463    1466#  1466    1468#  1468
              1470#  1470    1473#  1473    1475#  1475    1477#  1477    1564#  1564    1566#  1566    1568#  1568
              1571#  1571    1573#  1573    1575#  1575    1578#  1578    1580#  1580    1582#  1582    1749#  1749
              1751#  1751    1753#  1753    1758#  1758    1760#  1760    1762#  1762    1764#  1764    1769#  1769
              1771#  1771    1773#  1773    1775#  1775    1780#  1780    1782#  1782    1784#  1784    1786#  1786
              1788#  1788    1790#  1790    1792#  1792    1797#  1797    1799#  1799    1801#  1801    1803#  1803
              1805#  1805    1807#  1807    1809#  1809    1814#  1814    1816#  1816    1818#  1818    1820#  1820
              1822#  1822    1824#  1824    1826#  1826    1831#  1831    1833#  1833    1838#  1838    1840#  1840
              1842#  1842    1844#  1844    1846#  1846    1848#  1848    1850#  1850    1852#  1852    1854#  1854
              1859#  1859    1861#  1861    1863#  1863    1865#  1865    1867#  1867    1869#  1869    1871#  1671
              1873#  1873    1875#  1875    2026#  2026    2028#  2028    2030#  2030    2035#  2035    2037#  2037
              2039#  2039    2044#  2044    2046#  2046    2048#  2048    2050#  2050    2052#  2052    2054#  2054
              2059#  2059    2061#  2061    2063#  2063    2065#  2065    2067#  2067    2069#  2069    2071#  2071
              2077#  2077    2079#  2079    2081#  2081    2083#  2083    2085#  2085    2087#  2087    2089#  2089
              2091#  2091    2096#  2096    2098#  2098    2100#  2100    2102#  2102    2104#  2104    2106#  2106
              2108#  2108    2110#  2110    2112#  2112    2117#  2117    2119#  2119    2121#  2121    2123#  2123
```

```
              2125#  2125    2127#  2127    2129#  2129    2131#  2131    2133#  2133    2248#  2248    2250#  2250
```

```
2125#  2125   2127#  2127   2129#  2129   2131#  2131   2133#  2133   2248#  2248   2250#  2250
2252#  2252   2254#  2254   2256#  2256   2258#  2258   2260#  2260   2262#  2262   2264#  2264
2266#  2266   2268#  2268   2270#  2270   2272#  2272   2274#  2274   2276#  2276   2278#  2278
2280#  2280   2282#  2282   2284#  2284   2286#  2286   2288#  2288   2290#  2290   2292#  2292
2294#  2294   2296#  2296   2298#  2298   2300#  2300   2302#  2302   2304#  2304   2306#  2306
2308#  2308   2310#  2310   2312#  2312   2314#  2314   2316#  2316   2318#  2318   2320#  2320
2322#  2322   2327#  2327   2329#  2329   2331#  2331   2336#  2336   2338#  2338   2340#  2340
2342#  2342   2347#  2347   2349#  2349   2351#  2351   2356#  2356   2358#  2358   2360#  2360
2365#  2365   2367#  2367   2369#  2369   2374#  2374   2376#  2376   2378#  2378   2383#  2383
2385#  2385   2387#  2387   2392#  2392   2394#  2394   2396#  2396   2401#  2401   2403#  2403
2405#  2405   2410#  2410   2412#  2412   2414#  2414   2419#  2419   2421#  2421   2423#  2423
2428#  2428   2430#  2430   2432#  2432   2437#  2437   2439#  2439   2441#  2441   2446#  2446
2448#  2448   2450#  2450   2455#  2455   2457#  2457   2459#  2459   2464#  2464   2466#  2466
2468#  2468   2473#  2473   2475#  2475   2477#  2477   2482#  2482   2484#  2484   2486#  2486
2491#  2491   2493#  2493   2495#  2495   2500#  2500   2502#  2502   2504#  2504   2509#  2509
2511#  2511   2513#  2513   2518#  2518   2520#  2520   2522#  2522   2527#  2527   2529#  2529
2531#  2531   2536#  2536   2538#  2538   2540#  2540   2545#  2545   2547#  2547   2549#  2549
2554#  2554   2556#  2556   2558#  2558   2563#  2563   2565#  2565   2567#  2567   2572#  2572
2574#  2574   2576#  2576   2581#  2581   2583#  2583   2585#  2585   2590#  2590   2592#  2592
2594#  2594   2599#  2599   2601#  2601   2603#  2603   2608#  2608   2610#  2610   2612#  2612
2617#  2617   2619#  2619   2621#  2621   2626#  2626   2628#  2628   2630#  2630   2635#  2635
2637#  2637   2639#  2639   2644#  2644   2646#  2646   2648#  2648   2653#  2653   2655#  2655
2657#  2657   2659#  2659   2664#  2664   2666#  2666   2671#  2671   2673#  2673   2675#  2675
2680#  2680   2682#  2682   2684#  2684   2686#  2686   2688#  2688   2690#  2690   2692#  2692
2694#  2694   2696#  2696   2701#  2701   2703#  2703   2705#  2705   2707#  2707   2709#  2709
2711#  2711   2713#  2713   2715#  2715   2717#  2717   2848#  2848   2850#  2850   2852#  2852
2857#  2857   2859#  2859   2861#  2861   2863#  2863   2865#  2865   2867#  2867   2872#  2872
2874#  2874   2876#  2876   2878#  2878   2880#  2880   2882#  2882   2887#  2887   2889#  2889
2891#  2891   2893#  2893   2895#  2895   2897#  2897   2899#  2899   2904#  2904   2906#  2906
2908#  2908   2910#  2910   2912#  2912   2914#  2914   2916#  2916   2918#  2918   2920#  2920
2925#  2925   2927#  2927   2929#  2929   2931#  2931   2933#  2933   2935#  2935   2937#  2937
2939#  2939   2941#  2941   3054#  3054   3056#  3056   3058#  3058   3063#  3063   3065#  3065
3067#  3067   3069#  3069   3071#  3071   3073#  3073   3075#  3075   3077#  3077   3082#  3082
3084#  3084   3086#  3086   3088#  3088   3090#  3090   3092#  3092   3094#  3094   3096#  3096
3098#  3098   3103#  3103   3105#  3105   3107#  3107   3109#  3109   3111#  3111   3113#  3113
3115#  3115   3117#  3117   3119#  3119   3225#  3225   3227#  3227   3229#  3229   3231#  3231
3233#  3233   3235#  3235   3237#  3237   3239#  3239   3244#  3244   3246#  3246   3248#  3248
3250#  3250   3252#  3252   3254#  3254   3259#  3259   3261#  3261   3263#  3263   3265#  3265
3267#  3267   3269#  3269   3274#  3274   3276#  3276   3278#  3278   3280#  3280   3282#  3282
3284#  3284   3286#  3286   3288#  3288   3290#  3290   3295#  3295   3297#  3297   3299#  3299
3301#  3301   3303#  3303   3305#  3305   3307#  3307   3309#  3309   3311#  3311   3417#  3417
3419#  3419   3421#  3421   3423#  3423   3425#  3425   3427#  3427   3429#  3429   3431#  3431
3436#  3436   3438#  3438   3440#  3440   3442#  3442   3444#  3444   3446#  3446   3451#  3451
3453#  3453   3455#  3455   3457#  3457   3459#  3459   3461#  3461   3466#  3466   3468#  3468
3470#  3470   3472#  3472   3474#  3474   3476#  3476   3478#  3478   3480#  3480   3482#  3482
3487#  3487   3489#  3489   3491#  3491   3493#  3493   3495#  3495   3497#  3497   3499#  3499
3501#  3501   3503#  3503   3609#  3609   3611#  3611   3613#  3613   3615#  3615   3617#  3617
3619#  3619   3621#  3621   3623#  3623   3628#  3628   3630#  3630   3632#  3632   3634#  3634
3636#  3636   3638#  3638   3643#  3643   3645#  3645   3647#  3647   3649#  3649   3651#  3651
3653#  3653   3658#  3658   3660#  3660   3662#  3662   3664#  3664   3666#  3666   3668#  3668
3670#  3670   3672#  3672   3674#  3674   3679#  3679   3681#  3681   3683#  3683   3685#  3685
3687#  3687   3689#  3689   3691#  3691   3693#  3693   3695#  3695   3839#  3839   3841#  3841
3843#  3843   3848#  3848   3850#  3850   3852#  3852   3857#  3857   3859#  3859   3861#  3861
```

```
3863#  3863   3865#  3865   3870#  3870   3872#  3872   3874#  3874   3876#  3876   3878#  3878
```

```
1553
      3863#  3863   3865#  3865   3870#  3870   3872#  3872   3874#  3874   3876#  3876   3878#  3878
      3880#  3880   3882#  3882   3884#  3884   3886#  3886   3891#  3891   3893#  3893   3895#  3895      SEQ 1566
      3897#  3897   3899#  3899   3901#  3901   3903#  3903   3905#  3905   3907#  3907   4043#  4043
      4045#  4045   4047#  4047   4052#  4052   4054#  4054   4056#  4056   4058#  4058   4063#  4063
      4065#  4065   4067#  4067   4069#  4069   4074#  4074   4076#  4076   4078#  4078   4080#  4080
      4082#  4082   4084#  4084   4086#  4086   4088#  4088   4090#  4090   4095#  4095   4097#  4097
      4099#  4099   4101#  4101   4103#  4103   4105#  4105   4107#  4107   4109#  4109   4111#  4111
      4235#  4235   4237#  4237   4239#  4239   4244#  4244   4246#  4246   4248#  4248   4253#  4253
      4255#  4255   4257#  4257   4262#  4262   4264#  4264   4266#  4266   4271#  4271   4273#  4273
      4275#  4275   4277#  4277   4279#  4279   4281#  4281   4283#  4283   4285#  4285   4287#  4287
      4292#  4292   4294#  4294   4296#  4296   4298#  4298   4300#  4300   4302#  4302   4304#  4304
      4306#  4306   4308#  4308   4472#  4472   4474#  4474   4476#  4476   4482#  4482   4484#  4484
      4486#  4486   4488#  4488   4494#  4494   4496#  4496   4498#  4498   4500#  4500   4505#  4505
      4507#  4507   4509#  4509   4511#  4511   4513#  4513   4518#  4518   4520#  4520   4522#  4522
      4524#  4524   4526#  4526   4531#  4531   4533#  4533   4535#  4535   4540#  4540   4542#  4542
      4544#  4544   4546#  4546   4548#  4548   4550#  4550   4552#  4552   4554#  4554   4556#  4556
      4558#  4558   4560#  4560   4562#  4562   4564#  4564   4569#  4569   4571#  4571   4573#  4573
      4575#  4575   4577#  4577   4579#  4579   4581#  4581   4583#  4583   4585#  4585   4590#  4590
      4592#  4592   4594#  4594   4596#  4596   4598#  4598   4600#  4600   4602#  4602   4604#  4604
      4606#  4606   4766#  4766   4768#  4768   4770#  4770   4776#  4776   4778#  4778   4780#  4780
      4782#  4782   4784#  4784   4786#  4786   4788#  4788   4790#  4790   4792#  4792   4794#  4794
      4799#  4799   4801#  4801   4803#  4803   4805#  4805   4810#  4810   4812#  4812   4814#  4814
      4816#  4816   4818#  4818   4820#  4820   4822#  4822   4824#  4824   4826#  4826   4831#  4831
      4833#  4833   4835#  4835   4837#  4837   4839#  4839   4841#  4841   4843#  4843   4845#  4845
      4847#  4847   5048#  5048   5050#  5050   5052#  5052   5058#  5058   5060#  5060   5062#  5062
      5064#  5064   5066#  5066   5068#  5068   5070#  5070   5072#  5072   5079#  5079   5081#  5081
      5083#  5083   5085#  5085   5087#  5087   5089#  5089   5091#  5091   5093#  5093   5098#  5098
      5100#  5100   5102#  5102   5104#  5104   5110#  5110   5112#  5112   5114#  5114   5116#  5116
      5118#  5118   5124#  5124   5126#  5126   5128#  5128   5130#  5130   5132#  5132   5137#  5137
      5139#  5139   5141#  5141   5143#  5143   5145#  5145   5147#  5147   5149#  5149   5151#  5151
      5153#  5153   5158#  5158   5160#  5160   5162#  5162   5164#  5164   5166#  5166   5168#  5168
      5170#  5170   5172#  5172   5174#  5174   5324#  5324   5326#  5326   5328#  5328   5330#  5330
      5332#  5332   5337#  5337   5339#  5339   5341#  5341   5343#  5343   5345#  5345   5347#  5347
      5349#  5349   5351#  5351   5353#  5353   5355#  5355   5360#  5360   5362#  5362   5364#  5364
      5366#  5366   5368#  5368   5370#  5370   5372#  5372   5374#  5374   5379#  5379   5381#  5381
      5383#  5383   5385#  5385   5387#  5387   5389#  5389   5391#  5391   5393#  5393   5398#  5398
      5400#  5400   5402#  5402   5404#  5404   5406#  5406   5408#  5408   5413#  5413   5415#  5415
      5420#  5420   5422#  5422   5424#  5424   5426#  5426   5428#  5428   5430#  5430   5432#  5432
      5434#  5434   5436#  5436   5441#  5441   5443#  5443   5445#  5445   5447#  5447   5449#  5449
      5451#  5451   5453#  5453   5455#  5455   5457#  5457   5595#  5595   5597#  5597   5599#  5599
      5601#  5601   5606#  5606   5608#  5608   5610#  5610   5612#  5612   5614#  5614   5616#  5616
      5621#  5621   5623#  5623   5625#  5625   5627#  5627   5629#  5629   5631#  5631   5636#  5636
      5638#  5638   5640#  5640   5642#  5642   5644#  5644   5646#  5646   5648#  5648   5653#  5653
      5655#  5655   5657#  5657   5659#  5659   5661#  5661   5663#  5663   5665#  5665   5667#  5667
      5669#  5669   5674#  5674   5676#  5676   5678#  5678   5680#  5680   5682#  5682   5684#  5684
      5686#  5686   5688#  5688   5690#  5690   5811#  5811   5813#  5813   5815#  5815   5818#  5818
      5820#  5820   5822#  5822   5941#  5941   5946#  5946   5948#  5948   5950#  5950   5952#  5952
      5954#  5954   5956#  5956   5958#  5958   5960#  5960   5962#  5962   5964#  5964   5966#  5966
      5968#  5968   5970#  5970   5972#  5972   5974#  5974   5976#  5976   5978#  5978   5980#  5980
      5982#  5982   5984#  5984   5986#  5986   5988#  5988   5990#  5990   5992#  5992   5994#  5994
      5996#  5996   5998#  5998   6000#  6000   6002#  6002   6004#  6004   6006#  6006   6008#  6008
      6010#  6010   6012#  6012   6014#  6014   6016#  6016   6018#  6018   6020#  6020   6022#  6022
      6024#  6024   6026#  6026   6028#  6028   6030#  6030   6032#  6032   6034#  6034   6036#  6036
```

```
1554             6038#  6038   6040#  6040   6042#  6042   6044#  6044   6046#  6046   6048#  6048   6050#  6050      SEQ 1567    .LA
                 6052#  6052   6054#  6054   6056#  6056   6058#  6058   6060#  6060   6062#  6062   6064#  6064
                 6066#  6066   6068#  6068   6070#  6070   6072#  6072   6074#  6074   6076#  6076   6078#  6078
                 6080#  6080   6082#  6082   6084#  6084   6086#  6086   6088#  6088   6090#  6090   6092#  6092
                 6094#  6094   6096#  6096   6098#  6098   6100#  6100   6102#  6102   6104#  6104   6106#  6106
                 6108#  6108   6110#  6110   6112#  6112   6114#  6114   6116#  6116   6118#  6118   6120#  6120
                 6122#  6122   6124#  6124   6126#  6126   6128#  6128   6130#  6130   6132#  6132   6134#  6134
                 6136#  6136   6138#  6138   6140#  6140   6142#  6142   6144#  6144   6146#  6146   6148#  6148
                 6150#  6150   6152#  6152   6154#  6154   6156#  6156   6158#  6158   6160#  6160   6162#  6162
                 6164#  6164   6166#  6166   6168#  6168   6173#  6173   6181#  6181   6183#  6183   6185#  6185
                 6187#  6187   6189#  6189   6194#  6194   6196#  6196   6198#  6198   6200#  6200   6202#  6202
                 6204#  6204   6206#  6206   6208#  6208   6210#  6210   6212#  6212   6214#  6214   6216#  6216
                 6218#  6218   6220#  6220   6222#  6222   6224#  6224   6226#  6226   6228#  6228   6230#  6230
                 6232#  6232   6234#  6234   6236#  6236   6238#  6238   6240#  6240   6242#  6242   6244#  6244
                 6246#  6246   6248#  6248   6250#  6250   6252#  6252   6254#  6254   6256#  6256   6258#  6258
                 6263#  6263   6265#  6265   6267#  6267   6269#  6269   6272#  6272   6274#  6274   6276#  6276
                 6278#  6278   6281#  6281   6283#  6283   6285#  6285   6287#  6287   6290#  6290   6292#  6292
                 6294#  6294   6296#  6296   6299#  6299   6301#  6301   6303#  6303   6305#  6305   6308#  6308
                 6310#  6310   6312#  6312   6314#  6314   6317#  6317   6319#  6319   6321#  6321   6323#  6323
                 6326#  6326   6328#  6328   6330#  6330   6332#  6332   6335#  6335   6337#  6337   6339#  6339
                 6341#  6341   6344#  6344   6346#  6346   6348#  6348   6350#  6350   6353#  6353   6355#  6355
                 6357#  6357   6359#  6359   6362#  6362   6364#  6364   6366#  6366   6368#  6368   6371#  6371                  .LAD
                 6373#  6373   6375#  6375   6377#  6377   6380#  6380   6382#  6382   6384#  6384   6386#  6386
                 6389#  6389   6391#  6391   6393#  6393   6395#  6395   6398#  6398   6400#  6400   6402#  6402
                 6404#  6404   6407#  6407   6409#  6409   6411#  6411   6413#  6413   6416#  6416   6418#  6418
                 6420#  6420   6422#  6422   6425#  6425   6427#  6427   6429#  6429   6431#  6431   6434#  6434
                 6436#  6436   6438#  6438   6440#  6440   6443#  6443   6445#  6445   6447#  6447   6449#  6449
                 6452#  6452   6454#  6454   6456#  6456   6458#  6458   6461#  6461   6463#  6463   6465#  6465
                 6467#  6467   6470#  6470   6472#  6472   6474#  6474   6476#  6476   6479#  6479   6481#  6481
                 6483#  6483   6485#  6485   6488#  6488   6490#  6490   6492#  6492   6494#  6494   6497#  6497
                 6499#  6499   6501#  6501   6503#  6503   6506#  6506   6508#  6508   6510#  6510   6512#  6512
                 6515#  6515   6517#  6517   6519#  6519   6521#  6521   6524#  6524   6526#  6526   6528#  6528
                 6530#  6530   6533#  6533   6535#  6535   6537#  6537   6539#  6539   6542#  6542   6544#  6544
                 6546#  6546   6548#  6548   6551#  6551   6741#  6741   6743#  6743   6745#  6745   6747#  6747                  .LAN
                 6753#  6753   6755#  6755   6841#  6841   6843#  6843   6845#  6845   6847#  6847   6852#  6852
                 6854#  6854   6859#  6859   6861#  6861   6863#  6863   6865#  6865   6867#  6867   6961#  6961
                 6963#  6963   6965#  6965   6967#  6967   6969#  6969
     $GARG       1213#  1213   1215#  1215   1217#  1217   1220#  1220   1222#  1222   1224#  1224   1227#  1227
                 1229#  1229   1231#  1231   1234#  1234   1236#  1236   1238#  1238   1328#  1328   1330#  1330
                 1332#  1332   1424#  1424   1426#  1426   1428#  1428   1431#  1431   1433#  1433   1435#  1435
                 1438#  1438   1440#  1440   1442#  1442   1445#  1445   1447#  1447   1449#  1449   1452#  1452                  .LB
                 1454#  1454   1456#  1456   1459#  1459   1461#  1461   1463#  1463   1466#  1466   1468#  1468
                 1470#  1470   1473#  1473   1475#  1475   1477#  1477   1564#  1564   1566#  1566   1568#  1568
                 1571#  1571   1573#  1573   1575#  1575   1578#  1578   1580#  1580   1582#  1582   1749#  1749
                 1751#  1751   1753#  1753   1758#  1758   1760#  1760   1762#  1762   1764#  1764   1769#  1769
                 1771#  1771   1773#  1773   1775#  1775   1780#  1780   1782#  1782   1784#  1784   1786#  1786
                 1788#  1788   1790#  1790   1792#  1792   1797#  1797   1799#  1799   1801#  1801   1803#  1803
                 1805#  1805   1807#  1807   1809#  1809   1814#  1814   1816#  1816   1818#  1818   1820#  1820
                 1822#  1822   1824#  1824   1826#  1826   1831#  1831   1833#  1833   1838#  1838   1840#  1840
                 1842#  1842   1844#  1844   1846#  1846   1848#  1848   1850#  1850   1852#  1852   1854#  1854
                 1859#  1859   1861#  1861   1863#  1863   1865#  1865   1867#  1867   1869#  1869   1871#  1871
                 1873#  1873   1875#  1875   2026#  2026   2028#  2028   2030#  2030   2035#  2035   2037#  2037
                 2039#  2039   2044#  2044   2046#  2046   2048#  2048   2050#  2050   2052#  2052   2054#  2054

                 2059#  2059   2061#  2061   2063#  2063   2065#  2065   2067#  2067   2069#  2069   2071#  2071
```

```
2059#   2059    2061#   2061    2063#   2063    2065#   2065    2067#   2067    2069#   2069    2071#   2071
2077#   2077    2079#   2079    2081#   2081    2083#   2083    2085#   2085    2087#   2087    2089#   2089
2091#   2091    2096#   2096    2098#   2098    2100#   2100    2102#   2102    2104#   2104    2106#   2106
2108#   2108    2110#   2110    2112#   2112    2117#   2117    2119#   2119    2121#   2121    2123#   2123
2125#   2125    2127#   2127    2129#   2129    2131#   2131    2133#   2133    2248#   2248    2250#   2250
2252#   2252    2254#   2254    2256#   2256    2258#   2258    2260#   2260    2262#   2262    2264#   2264
2266#   2266    2268#   2268    2270#   2270    2272#   2272    2274#   2274    2276#   2276    2278#   2278
2280#   2280    2282#   2282    2284#   2284    2286#   2286    2288#   2288    2290#   2290    2292#   2292
2294#   2294    2296#   2296    2298#   2298    2300#   2300    2302#   2302    2304#   2304    2306#   2306
2308#   2308    2310#   2310    2312#   2312    2314#   2314    2316#   2316    2318#   2318    2320#   2320
2322#   2322    2327#   2327    2329#   2329    2331#   2331    2336#   2336    2338#   2338    2340#   2340
2342#   2342    2347#   2347    2349#   2349    2351#   2351    2356#   2356    2358#   2358    2360#   2360
2365#   2365    2367#   2367    2369#   2369    2374#   2374    2376#   2376    2378#   2378    2383#   2383
2385#   2385    2387#   2387    2392#   2392    2394#   2394    2396#   2396    2401#   2401    2403#   2403
2405#   2405    2410#   2410    2412#   2412    2414#   2414    2419#   2419    2421#   2421    2423#   2423
2428#   2428    2430#   2430    2432#   2432    2437#   2437    2439#   2439    2441#   2441    2446#   2446
2448#   2448    2450#   2450    2455#   2455    2457#   2457    2459#   2459    2464#   2464    2466#   2466
2468#   2468    2473#   2473    2475#   2475    2477#   2477    2482#   2482    2484#   2484    2486#   2486
2491#   2491    2493#   2493    2495#   2495    2500#   2500    2502#   2502    2504#   2504    2509#   2509
2511#   2511    2513#   2513    2518#   2518    2520#   2520    2522#   2522    2527#   2527    2529#   2529
2531#   2531    2536#   2536    2538#   2538    2540#   2540    2545#   2545    2547#   2547    2549#   2549
2554#   2554    2556#   2556    2558#   2558    2563#   2563    2565#   2565    2567#   2567    2572#   2572
2574#   2574    2576#   2576    2581#   2581    2583#   2583    2585#   2585    2590#   2590    2592#   2592
2594#   2594    2599#   2599    2601#   2601    2603#   2603    2608#   2608    2610#   2610    2612#   2612
2617#   2617    2619#   2619    2621#   2621    2626#   2626    2628#   2628    2630#   2630    2635#   2635
2637#   2637    2639#   2639    2644#   2644    2646#   2646    2648#   2648    2653#   2653    2655#   2655
2657#   2657    2659#   2659    2664#   2664    2666#   2666    2671#   2671    2673#   2673    2675#   2675
2680#   2680    2682#   2682    2684#   2684    2686#   2686    2688#   2688    2690#   2690    2692#   2692
2694#   2694    2696#   2696    2701#   2701    2703#   2703    2705#   2705    2707#   2707    2709#   2709
2711#   2711    2713#   2713    2715#   2715    2717#   2717    2848#   2848    2850#   2850    2852#   2852
2857#   2857    2859#   2859    2861#   2861    2863#   2863    2865#   2865    2867#   2867    2872#   2872
2874#   2874    2876#   2876    2878#   2878    2880#   2880    2882#   2882    2887#   2887    2889#   2889
2891#   2891    2893#   2893    2895#   2895    2897#   2897    2899#   2899    2904#   2904    2906#   2906
2908#   2908    2910#   2910    2912#   2912    2914#   2914    2916#   2916    2918#   2918    2920#   2920
2925#   2925    2927#   2927    2929#   2929    2931#   2931    2933#   2933    2935#   2935    2937#   2937
2939#   2939    2941#   2941    3054#   3054    3056#   3056    3058#   3058    3063#   3063    3065#   3065
3067#   3067    3069#   3069    3071#   3071    3073#   3073    3075#   3075    3077#   3077    3082#   3082
3084#   3084    3086#   3086    3088#   3088    3090#   3090    3092#   3092    3094#   3094    3096#   3096
3098#   3098    3103#   3103    3105#   3105    3107#   3107    3109#   3109    3111#   3111    3113#   3113
3115#   3115    3117#   3117    3119#   3119    3225#   3225    3227#   3227    3229#   3229    3231#   3231
3233#   3233    3235#   3235    3237#   3237    3239#   3239    3244#   3244    3246#   3246    3248#   3248
3250#   3250    3252#   3252    3254#   3254    3259#   3259    3261#   3261    3263#   3263    3265#   3265
3267#   3267    3269#   3269    3274#   3274    3276#   3276    3278#   3278    3280#   3280    3282#   3282
3284#   3284    3286#   3286    3288#   3288    3290#   3290    3295#   3295    3297#   3297    3299#   3299
3301#   3301    3303#   3303    3305#   3305    3307#   3307    3309#   3309    3311#   3311    3417#   3417
3419#   3419    3421#   3421    3423#   3423    3425#   3425    3427#   3427    3429#   3429    3431#   3431
3436#   3436    3438#   3438    3440#   3440    3442#   3442    3444#   3444    3446#   3446    3451#   3451
3453#   3453    3455#   3455    3457#   3457    3459#   3459    3461#   3461    3466#   3466    3468#   3468
3470#   3470    3472#   3472    3474#   3474    3476#   3476    3478#   3478    3480#   3480    3482#   3482
3487#   3487    3489#   3489    3491#   3491    3493#   3493    3495#   3495    3497#   3497    3499#   3499
3501#   3501    3503#   3503    3609#   3609    3611#   3611    3613#   3613    3615#   3615    3617#   3617
3619#   3619    3621#   3621    3623#   3623    3628#   3628    3630#   3630    3632#   3632    3634#   3634
3636#   3636    3638#   3638    3643#   3643    3645#   3645    3647#   3647    3649#   3649    3651#   3651
```

```
3653#  3653   3658#  3658   3660#  3660   3662#  3662   3664#  3664   3666#  3666   3668#  3668
3670#  3670   3672#  3672   3674#  3674   3679#  3679   3681#  3681   3683#  3683   3685#  3685
3687#  3687   3689#  3689   3691#  3691   3693#  3693   3695#  3695   3839#  3839   3841#  3841
3843#  3843   3848#  3848   3850#  3850   3852#  3852   3857#  3857   3859#  3859   3861#  3861
3863#  3863   3865#  3865   3870#  3870   3872#  3872   3874#  3874   3876#  3876   3878#  3878
3880#  3880   3882#  3882   3884#  3884   3886#  3886   3891#  3891   3893#  3893   3895#  3895
3897#  3897   3899#  3899   3901#  3901   3903#  3903   3905#  3905   3907#  3907   4043#  4043
4045#  4045   4047#  4047   4052#  4052   4054#  4054   4056#  4056   4058#  4058   4063#  4063
4065#  4065   4067#  4067   4069#  4069   4074#  4074   4076#  4076   4078#  4078   4080#  4080
4082#  4082   4084#  4084   4086#  4086   4088#  4088   4090#  4090   4095#  4095   4097#  4097
4099#  4099   4101#  4101   4103#  4103   4105#  4105   4107#  4107   4109#  4109   4111#  4111
4235#  4235   4237#  4237   4239#  4239   4244#  4244   4246#  4246   4248#  4248   4253#  4253
4255#  4255   4257#  4257   4262#  4262   4264#  4264   4266#  4266   4271#  4271   4273#  4273
4275#  4275   4277#  4277   4279#  4279   4281#  4281   4283#  4283   4285#  4285   4287#  4287
4292#  4292   4294#  4294   4296#  4296   4298#  4298   4300#  4300   4302#  4302   4304#  4304
4306#  4306   4308#  4308   4472#  4472   4474#  4474   4476#  4476   4482#  4482   4484#  4484
4486#  4486   4488#  4488   4494#  4494   4496#  4496   4498#  4498   4500#  4500   4505#  4505
4507#  4507   4509#  4509   4511#  4511   4513#  4513   4518#  4518   4520#  4520   4522#  4522
4524#  4524   4526#  4526   4531#  4531   4533#  4533   4535#  4535   4540#  4540   4542#  4542
4544#  4544   4546#  4546   4548#  4548   4550#  4550   4552#  4552   4554#  4554   4556#  4556
4558#  4558   4560#  4560   4562#  4562   4564#  4564   4569#  4569   4571#  4571   4573#  4573
4575#  4575   4577#  4577   4579#  4579   4581#  4581   4583#  4583   4585#  4585   4590#  4590
4592#  4592   4594#  4594   4596#  4596   4598#  4598   4600#  4600   4602#  4602   4604#  4604
4606#  4606   4766#  4766   4768#  4768   4770#  4770   4776#  4776   4778#  4778   5780#  4780
4782#  4782   4784#  4784   4786#  4786   4788#  4788   4790#  4790   4792#  4792   4794#  4794
4799#  4799   4801#  4801   4803#  4803   4805#  4805   4810#  4810   4812#  4812   4814#  4814
4816#  4816   4818#  4818   4820#  4820   4822#  4822   4824#  4824   4826#  4826   4831#  4831
4833#  4833   4835#  4835   4837#  4837   4839#  4839   4841#  4841   4843#  4843   4845#  4845
4847#  4847   5048#  5048   5050#  5050   5052#  5052   5058#  5058   5060#  5060   5062#  5062
5064#  5064   5066#  5066   5068#  5068   5070#  5070   5072#  5072   5079#  5079   5081#  5081
5083#  5083   5085#  5085   5087#  5087   5089#  5089   5091#  5091   5093#  5093   5098#  5098
5100#  5100   5102#  5102   5104#  5104   5110#  5110   5112#  5112   5114#  5114   5116#  5116
5118#  5118   5124#  5124   5126#  5126   5128#  5128   5130#  5130   5132#  5132   5137#  5137
5139#  5139   5141#  5141   5143#  5143   5145#  5145   5147#  5147   5149#  5149   5151#  5151
5153#  5153   5158#  5158   5160#  5160   5162#  5162   5164#  5164   5166#  5166   5168#  5168
5170#  5170   5172#  5172   5174#  5174   5324#  5324   5326#  5326   5328#  5328   5330#  5330
5332#  5332   5337#  5337   5339#  5339   5341#  5341   5343#  5343   5345#  5345   5347#  5347
5349#  5349   5351#  5351   5353#  5353   5355#  5355   5360#  5360   5362#  5362   5364#  5364
5366#  5366   5368#  5368   5370#  5370   5372#  5372   5374#  5374   5379#  5379   5381#  5381
5383#  5383   5385#  5385   5387#  5387   5389#  5389   5391#  5391   5393#  5393   5398#  5398
5400#  5400   5402#  5402   5404#  5404   5406#  5406   5408#  5408   5413#  5413   5415#  5415
5420#  5420   5422#  5422   5424#  5424   5426#  5426   5428#  5428   5430#  5430   5432#  5432
5434#  5434   5436#  5436   5441#  5441   5443#  5443   5445#  5445   5447#  5447   5449#  5449
5451#  5451   5453#  5453   5455#  5455   5457#  5457   5595#  5595   5597#  5597   5599#  5599
5601#  5601   5606#  5606   5608#  5608   5610#  5610   5612#  5612   5614#  5614   5616#  5616
5621#  5621   5623#  5623   5625#  5625   5627#  5627   5629#  5629   5631#  5631   5636#  5636
5638#  5638   5640#  5640   5642#  5642   5644#  5644   5646#  5646   5648#  5648   5653#  5653
5655#  5655   5657#  5657   5659#  5659   5661#  5661   5663#  5663   5665#  5665   5667#  5667
5669#  5669   5674#  5674   5676#  5676   5678#  5678   5680#  5680   5682#  5682   5684#  5684
5686#  5686   5688#  5688   5690#  5690   5811#  5811   5813#  5813   5815#  5815   5818#  5818
5820#  5820   5822#  5822   5941#  5941   5946#  5946   5948#  5948   5950#  5950   5952#  5952
5954#  5954   5956#  5956   5958#  5958   5960#  5960   5962#  5962   5964#  5964   5966#  5966
5968#  5968   5970#  5970   5972#  5972   5974#  5974   5976#  5976   5978#  5978   5980#  5980
```

```
          5982#  5982   5984#  5984   5986#  5986   5988#  5988   5990#  5990   5992#  5992   5994#  5994
          5996#  5996   5998#  5998   6000#  6000   6002#  6002   6004#  6004   6006#  6006   6008#  6008
          6010#  6010   6012#  6012   6014#  6014   6016#  6016   6018#  6018   6020#  6020   6022#  6022
          6024#  6024   6026#  6026   6028#  6028   6030#  6030   6032#  6032   6034#  6034   6036#  6036
          6038#  6038   6040#  6040   6042#  6042   6044#  6044   6046#  6046   6048#  6048   6050#  6050
          6052#  6052   6054#  6054   6056#  6056   6058#  6058   6060#  6060   6062#  6062   6064#  6064
          6066#  6066   6068#  6068   6070#  6070   6072#  6072   6074#  6074   6076#  6076   6078#  6078
          6080#  6080   6082#  6082   6084#  6084   6086#  6086   6088#  6088   6090#  6090   6092#  6092
          6094#  6094   6096#  6096   6098#  6098   6100#  6100   6102#  6102   6104#  6104   6106#  6106
          6108#  6108   6110#  6110   6112#  6112   6114#  6114   6116#  6116   6118#  6118   6120#  6120
          6122#  6122   6124#  6124   6126#  6126   6128#  6128   6130#  6130   6132#  6132   6134#  6134
          6136#  6136   6138#  6138   6140#  6140   6142#  6142   6144#  6144   6146#  6146   6148#  6148
          6150#  6150   6152#  6152   6154#  6154   6156#  6156   6158#  6158   6160#  6160   6162#  6162
          6164#  6164   6166#  6166   6168#  6168   6173#  6173   6181#  6181   6183#  6183   6185#  6185
          6187#  6187   6189#  6189   6194#  6194   6196#  6196   6198#  6198   6200#  6200   6202#  6202
          6204#  6204   6206#  6206   6208#  6208   6210#  6210   6212#  6212   6214#  6214   6216#  6216
          6218#  6218   6220#  6220   6222#  6222   6224#  6224   6226#  6226   6228#  6228   6230#  6230
          6232#  6232   6234#  6234   6236#  6236   6238#  6238   6240#  6240   6242#  6242   6244#  6244
          6246#  6246   6248#  6248   6250#  6250   6252#  6252   6254#  6254   6256#  6256   6258#  6258
          6263#  6263   6265#  6265   6267#  6267   6269#  6269   6272#  6272   6274#  6274   6276#  6276
          6278#  6278   6281#  6281   6283#  6283   6285#  6285   6287#  6287   6290#  6290   6292#  6292
          6294#  6294   6296#  6296   6299#  6299   6301#  6301   6303#  6303   6305#  6305   6308#  6308
          6310#  6310   6312#  6312   6314#  6314   6317#  6317   6319#  6319   6321#  6321   6323#  6323
          6326#  6326   6328#  6328   6330#  6330   6332#  6332   6335#  6335   6337#  6337   6339#  6339
          6341#  6341   6344#  6344   6346#  6346   6348#  6348   6350#  6350   6353#  6353   6355#  6355
          6357#  6357   6359#  6359   6362#  6362   6364#  6364   6366#  6366   6368#  6368   6371#  6371
          6373#  6373   6375#  6375   6377#  6377   6380#  6380   6382#  6382   6384#  6384   6386#  6386
          6389#  6389   6391#  6391   6393#  6393   6395#  6395   6398#  6398   6400#  6400   6402#  6402
          6404#  6404   6407#  6407   6409#  6409   6411#  6411   6413#  6413   6416#  6416   6418#  6418
          6420#  6420   6422#  6422   6425#  6425   6427#  6427   6429#  6429   6431#  6431   6434#  6434
          6436#  6436   6438#  6438   6440#  6440   6443#  6443   6445#  6445   6447#  6447   6449#  6449
          6452#  6452   6454#  6454   6456#  6456   6458#  6458   6461#  6461   6463#  6463   6465#  6465
          6467#  6467   6470#  6470   6472#  6472   6474#  6474   6476#  6476   6479#  6479   6481#  6481
          6483#  6483   6485#  6485   6488#  6488   6490#  6490   6492#  6492   6494#  6494   6497#  6497
          6499#  6499   6501#  6501   6503#  6503   6506#  6506   6508#  6508   6510#  6510   6512#  6512
          6515#  6515   6517#  6517   6519#  6519   6521#  6521   6524#  6524   6526#  6526   6528#  6528
          6530#  6530   6533#  6533   6535#  6535   6537#  6537   6539#  6539   6542#  6542   6544#  6544
          6546#  6546   6548#  6548   6551#  6551   6741#  6741   6743#  6743   6745#  6745   6747#  6747
          6753#  6753   6755#  6755   6841#  6841   6843#  6843   6845#  6845   6847#  6847   6852#  6852
          6854#  6854   6859#  6859   6861#  6861   6863#  6863   6865#  6865   6867#  6867   6961#  6961
          6963#  6963   6965#  6965   6967#  6967   6969#  6969
%ADDR     1213#  1213   1215   1215#  1217   1217#  1220   1220#  1222   1222#  1224   1224#  1227   1227#
          1229   1229#  1231   1231#  1234   1234#  1236   1236#  1238   1238#  1328#  1328   1330   1330#
          1332   1332#  1424#  1424   1426   1426#  1428   1428#  1431   1431#  1433   1433#  1435   1435#
          1438   1438#  1440   1440#  1442   1442#  1445   1445#  1447   1447#  1449   1449#  1452   1452#
          1454   1454#  1456   1456#  1459   1459#  1461   1461#  1463   1463#  1466   1466#  1468   1468#
          1470   1470#  1473   1473#  1475   1475#  1477   1477#  1564#  1564   1566   1566#  1568   1568#
          1571   1571#  1573   1573#  1575   1575#  1578   1578#  1580   1580#  1582   1582#  1749#  1749
          1751   1751#  1753   1753#  1758#  1758   1760   1760#  1762   1762#  1764   1764#  1769#  1769
          1771   1771#  1773   1773#  1775   1775#  1780#  1780   1782   1782#  1784   1784#  1786   1786#
          1788   1788#  1790   1790#  1792   1792#  1797#  1797   1799   1799#  1801   1801#  1803   1803#
          1805   1805#  1807   1807#  1809   1809#  1814#  1814   1816   1816#  1818   1818#  1820   1820#
          1822   1822#  1824   1824#  1826   1826#  1831#  1831   1833   1833#  1838#  1838   1840   1840#
```

Q 1558

SEQ 1571

```
1842    1842#   1844    1844#   1846    1846#   1848    1848#   1850    1850#   1852    1852#   1854    1854#
1859#   1859    1861    1861#   1863    1863#   1865    1865#   1867    1867#   1869    1869#   1871    1871#
1873    1873#   1875    1875#   2026#   2026    2028    2028#   2030    2030#   2035#   2035    2037    2037#
2039    2039#   2044#   2044    2046    2046#   2048    2048#   2050    2050#   2052    2052#   2054    2054#
2059#   2059    2061    2061#   2063    2063#   2065    2065#   2067    2067#   2069    2069#   2071    2071#
2077#   2077    2079    2079#   2081    2081#   2083    2083#   2085    2085#   2087    2087#   2089    2089#
2091    2091#   2096#   2096    2098    2098#   2100    2100#   2102    2102#   2104    2104#   2106    2106#
2108    2108#   2110    2110#   2112    2112#   2117#   2117    2119    2119#   2121    2121#   2123    2123#
2125    2125#   2127    2127#   2129    2129#   2131    2131#   2133    2133#   2248#   2248    2250    2250#
2252    2252#   2254    2254#   2256    2256#   2258    2258#   2260    2260#   2262    2262#   2264    2264#
2266    2266#   2268    2268#   2270    2270#   2272    2272#   2274    2274#   2276    2276#   2278    2278#
2280    2280#   2282    2282#   2284    2284#   2286    2286#   2288    2288#   2290    2290#   2292    2292#
2294    2294#   2296    2296#   2298    2298#   2300    2300#   2302    2302#   2304    2304#   2306    2306#
2308    2308#   2310    2310#   2312    2312#   2314    2314#   2316    2316#   2318    2318#   2320    2320#
2322    2322#   2327#   2327    2329    2329#   2331    2331#   2336#   2336    2338    2338#   2340    2340#
2342    2342#   2347#   2347    2349    2349#   2351    2351#   2356#   2356    2358    2358#   2360    2360#
2365#   2365    2367    2367#   2369    2369#   2374#   2374    2376    2375#   2378    2378#   2383#   2383
2385    2385#   2387    2387#   2392#   2392    2394    2394#   2396    2396#   2401#   2401    2403    2403#
2405    2405#   2410#   2410    2412    2412#   2414    2414#   2419#   2419    2421    2421#   2423    2423#
2428#   2428    2430    2430#   2432    2432#   2437#   2437    2439    2439#   2441    2441#   2446#   2446
2448    2448#   2450    2450#   2455#   2455    2457    2457#   2459    2459#   2464#   2464    2466    2466#
2468    2468#   2473#   2473    2475    2475#   2477    2477#   2482#   2482    2484    2484#   2486    2486#
2491#   2491    2493    2493#   2495    2495#   2500#   2500    2502    2502#   2504    2504#   2509#   2509
2511    2511#   2513    2513#   2518#   2518    2520    2520#   2522    2522#   2527#   2527    2529    2529#
2531    2531#   2536#   2536    2538    2538#   2540    2540#   2545#   2545    2547    2547#   2549    2549#
2554#   2554    2556    2556#   2558    2558#   2563#   2563    2565    2565#   2567    2567#   2572#   2572
2574    2574#   2576    2576#   2581#   2581    2583    2583#   2585    2585#   2590#   2590    2592    2592#
2594    2594#   2599#   2599    2601    2601#   2603    2603#   2608#   2608    2610    2610#   2612    2612#
2617#   2617    2619    2619#   2621    2621#   2626#   2626    2628    2628#   2630    2630#   2635#   2635
2637    2637#   2639    2639#   2644#   2644    2646    2646#   2648    2648#   2653#   2653    2655    2655#
2657    2657#   2659    2659#   2664#   2664    2666    2666#   2671#   2671    2673    2673#   2675    2675#
2680#   2680    2682    2682#   2684    2684#   2686    2686#   2688    2688#   2690    2690#   2692    2692#
2694    2694#   2696    2696#   2701#   2701    2703    2703#   2705    2705#   2707    2707#   2709    2709#
2711    2711#   2713    2713#   2715    2715#   2717    2717#   2848#   2848    2850    2850#   2852    2852#
2857#   2857    2859    2859#   2861    2861#   2863    2863#   2865    2865#   2867    2867#   2872#   2872
2874    2874#   2876    2876#   2878    2878#   2880    2880#   2882    2882#   2887#   2887    2889    2889#
2891    2891#   2893    2893#   2895    2895#   2897    2897#   2899    2899#   2904#   2904    2906    2906#
2908    2908#   2910    2910#   2912    2912#   2914    2914#   2916    2916#   2918    2918#   2920    2920#
2925#   2925    2927    2927#   2929    2929#   2931    2931#   2933    2933#   2935    2935#   2937    2937#
2939    2939#   2941    2941#   3054#   3054    3056    3056#   3058    3058#   3063#   3063    3065    3065#
3067    3067#   3069    3069#   3071    3071#   3073    3073#   3075    3075#   3077    3077#   3082#   3082
3084    3084#   3086    3086#   3088    3088#   3090    3090#   3092    3092#   3094    3094#   3096    3096#
3098    3098#   3103#   3103    3105    3105#   3107    3107#   3109    3109#   3111    3111#   3113    3113#
3115    3115#   3117    3117#   3119    3119#   3225#   3225    3227#   3227    3229    3229#   3231    3231#
3233    3233#   3235    3235#   3237    3237#   3239    3239#   3244#   3244    3246    3246#   3248    3248#
3250    3250#   3252    3252#   3254    3254#   3259#   3259    3261    3261#   3263    3263#   3265    3265#
3267    3267#   3269    3269#   3274#   3274    3276    3276#   3278    3278#   3280    3280#   3282    3282#
3284    3284#   3286    3286#   3288    3288#   3290    3290#   3295#   3295    3297    3297#   3299    3299#
3301    3301#   3303    3303#   3305    3305#   3307    3307#   3309    3309#   3311    3311#   3417#   3417
3419#   3419    3421    3421#   3423    3423#   3425    3425#   3427    3427#   3429    3429#   3431    3431#
3436#   3436    3438    3438#   3440    3440#   3442    3442#   3444    3444#   3446    3445#   3451#   3451
3453    3453#   3455    3455#   3457    3457#   3459    3459#   3461    3461#   3466#   3466    3468    3468#
3470    3470#   3472    3472#   3474    3474#   3476    3476#   3478    3478#   3480    3480#   3482    3482#
```

```
3487#      3487       3489       3489#      3491       3491#      3493       3493#      3495       3495#      3497       3497#      3499       3499#
3501       3501#      3503       3503#      3609#      3609       3611#      3611       3613       3613#      3615       3615#      3617       3617#
3619       3619#      3621       3621#      3623       3623#      3628#      3628       3630       3630#      3632       3632#      3634       3634#
3636       3636#      3638       3638#      3643#      3643       3645       3645#      3647       3647#      3649       3649#      3651       3651#
3653       3653#      3658#      3658       3660       3660#      3662       3662#      3664       3664#      3666       3666#      3668       3668#
3670       3670#      3672       3672#      3674       3674#      3679#      3679       3681       3681#      3683       3683#      3685       3685#
3687       3687#      3689       3689#      3691       3691#      3693       3693#      3695       3695#      3839#      3839       3841       3841#
3843       3843#      3848#      3848       3850       3850#      3852       3852#      3857#      3857       3859       3859#      3861       3861#
3863       3863#      3865       3865#      3870#      3870       3872       3872#      3874       3874#      3876       3876#      3878       3878#
3880       3880#      3882       3882#      3884       3884#      3886       3886#      3891#      3891       3893       3893#      3895       3895#
3897       3897#      3899       3899#      3901       3901#      3903       3903#      3905       3905#      3907       3907#      4043#      4043
4045       4045#      4047       4047#      4052#      4052       4054       4054#      4056       4056#      4058       4058#      4063#      4063
4065       4065#      4067       4067#      4069       4069#      4074#      4074       4076       4076#      4078       4078#      4080       4080#
4082       4082#      4084       4084#      4086       4086#      4088       4088#      4090       4090#      4095#      4095       4097       4097#
4099       4099#      4101       4101#      4103       4103#      4105       4105#      4107       4107#      4109       4109#      4111       4111#
4235#      4235       4237       4237#      4239       4239#      4244#      4244       4246       4246#      4248       4248#      4253#      4253
4255       4255#      4257       4257#      4262#      4262       4264       4264#      4266       4266#      4271#      4271       4273       4273#
4275       4275#      4277       4277#      4279       4279#      4281       4281#      4283       4283#      4285       4285#      4287       4287#
4292#      4292       4294       4294#      4296       4296#      4298       4298#      4300       4300#      4302       4302#      4304       4304#
4306       4306#      4308       4308#      4472#      4472       4474       4474#      4476       4476#      4482#      4482       4484       4484#
4486       4486#      4488       4488#      4494#      4494       4496       4496#      4498       4498#      4500       4500#      4505#      4505
4507       4507#      4509       4509#      4511       4511#      4513       4513#      4518#      4518       4520       4520#      4522       4522#
4524       4524#      4526       4526#      4531#      4531       4533       4533#      4535       4535#      4540#      4540       4542       4542#
4544       4544#      4546       4546#      4548       4548#      4550       4550#      4552       4552#      4554       4554#      4556       4556#
4558       4558#      4560       4560#      4562       4562#      4564       4564#      4569#      4569       4571       4571#      4573       4573#
4575       4575#      4577       4577#      4579       4579#      4581       4581#      4583       4583#      4585       4585#      4590#      4590
4592       4592#      4594       4594#      4596       4596#      4598       4598#      4600       4600#      4602       4602#      4604       4604#
4606       4606#      4766#      4766       4768       4768#      4770       4770#      4776#      4776       4778       4778#      4780       4780#
4782       4782#      4784       4784#      4786       4786#      4788       4788#      4790       4790#      4792       4792#      4794       4794#
4799#      4799       4801       4801#      4803       4803#      4805       4805#      4810#      4810       4812       4812#      4814       4814#
4816       4816#      4818       4818#      4820       4820#      4822       4822#      4824       4824#      4826       4826#      4831#      4831
4833       4833#      4835       4835#      4837       4837#      4839       4839#      4841       4841#      4843       4843#      4845       4845#
4847       4847#      5048#      5048       5050       5050#      5052       5052#      5058#      5058       5060       5060#      5062       5062#
5064       5064#      5066       5066#      5068       5068#      5070       5070#      5072       5072#      5079#      5079       5081       5081#
5083       5083#      5085       5085#      5087       5087#      5089       5089#      5091       5091#      5093       5093#      5098#      5098
5100       5100#      5102       5102#      5104       5104#      5110#      5110       5112       5112#      5114       5114#      5116       5116#
5118       5118#      5124#      5124       5126       5126#      5128       5128#      5130       5130#      5132       5132#      5137#      5137
5139       5139#      5141       5141#      5143       5143#      5145       5145#      5147       5147#      5149       5149#      5151       5151#
5153       5153#      5158#      5158       5160       5160#      5162       5162#      5164       5164#      5166       5166#      5168       5168#
5170       5170#      5172       5172#      5174       5174#      5324#      5324       5326       5326#      5328       5328#      5330       5330#
5332       5332#      5337#      5337       5339       5339#      5341       5341#      5343       5343#      5345       5345#      5347       5347#
5349       5349#      5351       5351#      5353       5353#      5355       5355#      5360#      5360       5362       5362#      5364       5364#
5366       5366#      5368       5368#      5370       5370#      5372       5372#      5374       5374#      5379#      5379       5381       5381#
5383       5383#      5385       5385#      5387       5387#      5389       5389#      5391       5391#      5393       5393#      5398#      5398
5400       5400#      5402       5402#      5404       5404#      5406       5406#      5408       5408#      5413#      5413       5415       5415#
5420#      5420       5422       5422#      5424       5424#      5426       5426#      5428       5428#      5430       5430#      5432       5432#
5434       5434#      5436       5436#      5441#      5441       5443       5443#      5445       5445#      5447       5447#      5449       5449#
5451       5451#      5453       5453#      5455       5455#      5457       5457#      5595#      5595       5597       5597#      5599       5599#
5601       5601#      5606#      5606       5608       5608#      5610       5610#      5612       5612#      5614       5614#      5616       5616#
5621#      5621       5623       5623#      5625       5625#      5627       5627#      5629       5629#      5631       5631#      5636#      5636
5638       5638#      5640       5640#      5642       5642#      5644       5644#      5646       5646#      5648       5648#      5653#      5653
5655       5655#      5657       5657#      5659       5659#      5661       5661#      5663       5663#      5665       5665#      5667       5667#
5669       5669#      5674#      5674       5676       5676#      5678       5678#      5680       5680#      5682       5682#      5684       5684#
```

```
EQ 1560 5686   5686#  5688   5688#  5690   5690#  5811#  5811   5813   5813#  5815   5815#  5818   5818#
        5820   5820#  5822   5822#  5941#  5941   5946#  5946   5948   5948#  5950   5950#  5952   5952#   SEQ 1573
        5954   5954#  5956   5956#  5958   5958#  5960   5960#  5962   5962#  5964   5964#  5966   5966#
        5968   5968#  5970   5970#  5972   5972#  5974   5974#  5976   5976#  5978   5978#  5980   5980#
        5982   5982#  5984   5984#  5986   5986#  5988   5988#  5990   5990#  5992   5992#  5994   5994#
        5996   5996#  5998   5998#  6000   6000#  6002   6002#  6004   6004#  6006   6006#  6008   6008#
        6010   6010#  6012   6012#  6014   6014#  6016   6016#  6018   6018#  6020   6020#  6022   6022#
        6024   6024#  6026   6026#  6028   6028#  6030   6030#  6032   6032#  6034   6034#  6036   6036#
        6038   6038#  6040   6040#  6042   6042#  6044   6044#  6046   6046#  6048   6048#  6050   6050#
        6052   6052#  6054   6054#  6056   6056#  6058   6058#  6060   6060#  6062   6062#  6064   6064#
        6066   6066#  6068   6068#  6070   6070#  6072   6072#  6074   6074#  6076   6076#  6078   6078#
        6080   6080#  6082   6082#  6084   6084#  6086   6086#  6088   6088#  6090   6090#  6092   6092#
        6094   6094#  6096   6096#  6098   6098#  6100   6100#  6102   6102#  6104   6104#  6106   6106#
        6108   6108#  6110   6110#  6112   6112#  6114   6114#  6116   6116#  6118   6118#  6120   6120#
        6122   6122#  6124   6124#  6126   6126#  6128   6128#  6130   6130#  6132   6132#  6134   6134#
        6136   6136#  6138   6138#  6140   6140#  6142   6142#  6144   6144#  6146   6146#  6148   6148#
        6150   6150#  6152   6152#  6154   6154#  6156   6156#  6158   6158#  6160   6160#  6162   6162#
        6164   6164#  6166   6166#  6168   6168#  6173#  6173   6181#  6181   6183   6183#  6185   6185#
        6187   6187#  6189   6189#  6194#  6194   6196   6196#  6198   6198#  6200   6200#  6202   6202#
        6204   6204#  6206   6206#  6208   6208#  6210   6210#  6212   6212#  6214   6214#  6216   6216#
        6218   6218#  6220   6220#  6222   6222#  6224   6224#  6226   6226#  6228   6228#  6230   6230#
        6232   6232#  6234   6234#  6236   6236#  6238   6238#  6240   6240#  6242   6242#  6244   6244#
        6246   6246#  6248   6248#  6250   6250#  6252   6252#  6254   6254#  6256   6256#  6258   6258#
        6263#  6263   6265   6265#  6267   6267#  6269   6269#  6272   6272#  6274   6274#  6276   6276#
        6278   6278#  6281   6281#  6283   6283#  6285   6285#  6287   6287#  6290   6290#  6292   6292#
        6294   6294#  6296   6296#  6299   6299#  6301   6301#  6303   6303#  6305   6305#  6308   6308#
        6310   6310#  6312   6312#  6314   6314#  6317   6317#  6319   6319#  6321   6321#  6323   6323#
        6326   6326#  6328   6328#  6330   6330#  6332   6332#  6335   6335#  6337   6337#  6339   6339#
        6341   6341#  6344   6344#  6346   6346#  6348   6348#  6350   6350#  6353   6353#  6355   6355#
        6357   6357#  6359   6359#  6362   6362#  6364   6364#  6366   6366#  6368   6368#  6371   6371#
        6373   6373#  6375   6375#  6377   6377#  6380   6380#  6382   6382#  6384   6384#  6386   6386#
        6389   6389#  6391   6391#  6393   6393#  6395   6395#  6398   6398#  6400   6400#  6402   6402#
        6404   6404#  6407   6407#  6409   6409#  6411   6411#  6413   6413#  6416   6416#  6418   6418#
        6420   6420#  6422   6422#  6425   6425#  6427   6427#  6429   6429#  6431   6431#  6434   6434#
        6436   6436#  6438   6438#  6440   6440#  6443   6443#  6445   6445#  6447   6447#  6449   6449#
        6452   6452#  6454   6454#  6456   6456#  6458   6458#  6461   6461#  6463   6463#  6465   6465#
        6467   6467#  6470   6470#  6472   6472#  6474   6474#  6476   6476#  6479   6479#  6481   6481#
        6483   6483#  6485   6485#  6488   6488#  6490   6490#  6492   6492#  6494   6494#  6497   6497#
        6499   6499#  6501   6501#  6503   6503#  6506   6506#  6508   6508#  6510   6510#  6512   6512#
        6515   6515#  6517   6517#  6519   6519#  6521   6521#  6524   6524#  6526   6526#  6528   6528#
        6530   6530#  6533   6533#  6535   6535#  6537   6537#  6539   6539#  6542   6542#  6544   6544#
        6546   6546#  6548   6548#  6551   6551#  6741#  6741   6743   6743#  6745#  6745   6747   6747#
        6753#  6753   6755#  6755   6841#  6841   6843   6843#  6845   6845#  6847   6847#  6852   6852#
        6854   6854#  6859   6859#  6861   6861#  6863   6863#  6865   6865#  6867   6867#  6961#  6961
        6963   6963#  6965   6965#  6967   6967#  6969   6969#
%ML     1213#  1213   1215#  1215   1217#  1217   1220#  1220   1222#  1222   1224#  1224   1227#  1227
        1229#  1229   1231#  1231   1234#  1234   1236#  1236   1238#  1238   1328#  1328   1330#  1330
        1332#  1332   1424#  1424   1426#  1426   1428#  1428   1431#  1431   1433#  1433   1435#  1435
        1438#  1438   1440#  1440   1442#  1442   1445#  1445   1447#  1447   1449#  1449   1452#  1452
        1454#  1454   1456#  1456   1459#  1459   1461#  1461   1463#  1463   1466#  1466   1468#  1468
        1470#  1470   1473#  1473   1475#  1475   1477#  1477   1564#  1564   1566#  1566   1568#  1568
        1571#  1571   1573#  1573   1575#  1575   1578#  1578   1580#  1580   1582#  1582   1749#  1749
        1751#  1751   1753#  1753   1758#  1758   1760#  1760   1762#  1762   1764#  1764   1769#  1769
```

```
1771#  1771   1773#  1773   1775#  1775   1780#  1780   1782#  1782   1784#  1784   1786#  1786
1788#  1788   1790#  1790   1792#  1792   1797#  1797   1799#  1799   1801#  1801   1803#  1803     SEQ 1574
1805#  1805   1807#  1807   1809#  1809   1814#  1814   1816#  1816   1818#  1818   1820#  1820
1822#  1822   1824#  1824   1826#  1826   1831#  1831   1833#  1833   1838#  1838   1840#  1840
1842#  1842   1844#  1844   1846#  1846   1848#  1848   1850#  1850   1852#  1852   1854#  1854
1859#  1859   1861#  1861   1863#  1863   1865#  1865   1867#  1867   1869#  1869   1871#  1871
1873#  1873   1875#  1875   2026#  2026   2028#  2028   2030#  2030   2035#  2035   2037#  2037
2039#  2039   2044#  2044   2046#  2046   2048#  2048   2050#  2050   2052#  2052   2054#  2054
2059#  2059   2061#  2061   2063#  2063   2065#  2065   2067#  2067   2069#  2069   2071#  2071
2077#  2077   2079#  2079   2081#  2081   2083#  2083   2085#  2085   2087#  2087   2089#  2089
2091#  2091   2096#  2096   2098#  2098   2100#  2100   2102#  2102   2104#  2104   2106#  2106
2108#  2108   2110#  2110   2112#  2112   2117#  2117   2119#  2119   2121#  2121   2123#  2123
2125#  2125   2127#  2127   2129#  2129   2131#  2131   2133#  2133   2248#  2248   2250#  2250
2252#  2252   2254#  2254   2256#  2256   2258#  2258   2260#  2260   2262#  2262   2264#  2264
2266#  2266   2268#  2268   2270#  2270   2272#  2272   2274#  2274   2276#  2276   2278#  2278
2280#  2280   2282#  2282   2284#  2284   2286#  2286   2288#  2288   2290#  2290   2292#  2292
2294#  2294   2296#  2296   2298#  2298   2300#  2300   2302#  2302   2304#  2304   2306#  2306
2308#  2308   2310#  2310   2312#  2312   2314#  2314   2316#  2316   2318#  2318   2320#  2320
2322#  2322   2327#  2327   2329#  2329   2331#  2331   2336#  2336   2338#  2338   2340#  2340
2342#  2342   2347#  2347   2349#  2349   2351#  2351   2356#  2356   2358#  2358   2360#  2360
2365#  2365   2367#  2367   2369#  2369   2374#  2374   2376#  2376   2378#  2378   2383#  2383
2385#  2385   2387#  2387   2392#  2392   2394#  2394   2396#  2396   2401#  2401   2403#  2403
2405#  2405   2410#  2410   2412#  2412   2414#  2414   2419#  2419   2421#  2421   2423#  2423
2428#  2428   2430#  2430   2432#  2432   2437#  2437   2439#  2439   2441#  2441   2446#  2446
2448#  2448   2450#  2450   2455#  2455   2457#  2457   2459#  2459   2464#  2464   2466#  2466
2468#  2468   2473#  2473   2475#  2475   2477#  2477   2482#  2482   2484#  2484   2486#  2486
2491#  2491   2493#  2493   2495#  2495   2500#  2500   2502#  2502   2504#  2504   2509#  2509
2511#  2511   2513#  2513   2518#  2518   2520#  2520   2522#  2522   2527#  2527   2529#  2529
2531#  2531   2536#  2536   2538#  2538   2540#  2540   2545#  2545   2547#  2547   2549#  2549
2554#  2554   2556#  2556   2558#  2558   2563#  2563   2565#  2565   2567#  2567   2572#  2572
2574#  2574   2576#  2576   2581#  2581   2583#  2583   2585#  2585   2590#  2590   2592#  2592
2594#  2594   2599#  2599   2601#  2601   2603#  2603   2608#  2608   2610#  2610   2612#  2612
2617#  2617   2619#  2619   2621#  2621   2626#  2626   2628#  2628   2630#  2630   2635#  2635
2637#  2637   2639#  2639   2644#  2644   2646#  2646   2648#  2648   2653#  2653   2655#  2655
2657#  2657   2659#  2659   2664#  2664   2666#  2666   2671#  2671   2673#  2673   2675#  2675
2680#  2680   2682#  2682   2684#  2684   2686#  2686   2688#  2688   2690#  2690   2692#  2692
2694#  2694   2696#  2696   2701#  2701   2703#  2703   2705#  2705   2707#  2707   2709#  2709
2711#  2711   2713#  2713   2715#  2715   2717#  2717   2848#  2848   2850#  2850   2852#  2852
2857#  2857   2859#  2859   2861#  2861   2863#  2863   2865#  2865   2867#  2867   2872#  2872
2874#  2874   2876#  2876   2878#  2878   2880#  2880   2882#  2882   2887#  2887   2889#  2889
2891#  2891   2893#  2893   2895#  2895   2897#  2897   2899#  2899   2904#  2904   2906#  2906
2908#  2908   2910#  2910   2912#  2912   2914#  2914   2916#  2916   2918#  2918   2920#  2920
2925#  2925   2927#  2927   2929#  2929   2931#  2931   2933#  2933   2935#  2935   2937#  2937
2939#  2939   2941#  2941   3054#  3054   3056#  3056   3058#  3058   3063#  3063   3065#  3065
3067#  3067   3069#  3069   3071#  3071   3073#  3073   3075#  3075   3077#  3077   3082#  3082
3084#  3084   3086#  3086   3088#  3088   3090#  3090   3092#  3092   3094#  3094   3096#  3096
3098#  3098   3103#  3103   3105#  3105   3107#  3107   3109#  3109   3111#  3111   3113#  3113
3115#  3115   3117#  3117   3119#  3119   3225#  3225   3227#  3227   3229#  3229   3231#  3231
3233#  3233   3235#  3235   3237#  3237   3239#  3239   3244#  3244   3246#  3246   3248#  3248
3250#  3250   3252#  3252   3254#  3254   3259#  3259   3261#  3261   3263#  3263   3265#  3265
3267#  3267   3269#  3269   3274#  3274   3276#  3276   3278#  3278   3280#  3280   3282#  3282
3284#  3284   3286#  3286   3288#  3288   3290#  3290   3295#  3295   3297#  3297   3299#  3299
3301#  3301   3303#  3303   3305#  3305   3307#  3307   3309#  3309   3311#  3311   3417#  3417
```

```
3419#  3419   3421#  3421   3423#  3423   3425#  3425   3427#  3427   3429#  3429   3431#  3431
3436#  3436   3438#  3438   3440#  3440   3442#  3442   3444#  3444   3446#  3446   3451#  3451     SEQ 1575
```

```
3419#  3419   3421#  3421   3423#  3423   3425#  3425   3427#  3427   3429#  3429   3431#  3431        SEQ 1575
3436#  3436   3438#  3438   3440#  3440   3442#  3442   3444#  3444   3446#  3446   3451#  3451
3453#  3453   3455#  3455   3457#  3457   3459#  3459   3461#  3461   3466#  3466   3468#  3468
3470#  3470   3472#  3472   3474#  3474   3476#  3476   3478#  3478   3480#  3480   3482#  3482
3487#  3487   3489#  3489   3491#  3491   3493#  3493   3495#  3495   3497#  3497   3499#  3499
3501#  3501   3503#  3503   3609#  3609   3611#  3611   3613#  3613   3615#  3615   3617#  3617
3619#  3619   3621#  3621   3623#  3623   3628#  3628   3630#  3630   3632#  3632   3634#  3634
3636#  3636   3638#  3638   3643#  3643   3645#  3645   3647#  3647   3649#  3649   3651#  3651
3653#  3653   3658#  3658   3660#  3660   3662#  3662   3664#  3664   3666#  3666   3668#  3668
3670#  3670   3672#  3672   3674#  3674   3679#  3679   3681#  3681   3683#  3683   3685#  3685
3687#  3687   3689#  3689   3691#  3691   3693#  3693   3695#  3695   3839#  3839   3841#  3841
3843#  3843   3848#  3848   3850#  3850   3852#  3852   3857#  3857   3859#  3859   3861#  3861
3863#  3863   3865#  3865   3870#  3870   3872#  3872   3874#  3874   3876#  3876   3878#  3878
3880#  3880   3882#  3882   3884#  3884   3886#  3886   3891#  3891   3893#  3893   3895#  3895
3897#  3897   3899#  3899   3901#  3901   3903#  3903   3905#  3905   3907#  3907   4043#  4043
4045#  4045   4047#  4047   4052#  4052   4054#  4054   4056#  4056   4058#  4058   4063#  4063
4065#  4065   4067#  4067   4069#  4069   4074#  4074   4076#  4076   4078#  4078   4080#  4080
4082#  4082   4084#  4084   4086#  4086   4088#  4088   4090#  4090   4095#  4095   4097#  4097
4099#  4099   4101#  4101   4103#  4103   4105#  4105   4107#  4107   4109#  4109   4111#  4111
4235#  4235   4237#  4237   4239#  4239   4244#  4244   4246#  4246   4248#  4248   4253#  4253
4255#  4255   4257#  4257   4262#  4262   4264#  4264   4266#  4266   4271#  4271   4273#  4273
4275#  4275   4277#  4277   4279#  4279   4281#  4281   4283#  4283   4285#  4285   4287#  4287
4292#  4292   4294#  4294   4296#  4296   4298#  4298   4300#  4300   4302#  4302   4304#  4304
4306#  4306   4308#  4308   4472#  4472   4474#  4474   4476#  4476   4482#  4482   4484#  4484
4486#  4486   4488#  4488   4494#  4494   4496#  4496   4498#  4498   4500#  4500   4505#  4505
4507#  4507   4509#  4509   4511#  4511   4513#  4513   4518#  4518   4520#  4520   4522#  4522
4524#  4524   4526#  4526   4531#  4531   4533#  4533   4535#  4535   4540#  4540   4542#  4542
4544#  4544   4546#  4546   4548#  4548   4550#  4550   4552#  4552   4554#  4554   4556#  4556
4558#  4558   4560#  4560   4562#  4562   4564#  4564   4569#  4569   4571#  4571   4573#  4573
4575#  4575   4577#  4577   4579#  4579   4581#  4581   4583#  4583   4585#  4585   4590#  4590
4592#  4592   4594#  4594   4596#  4596   4598#  4598   4600#  4600   4602#  4602   4604#  4604
4606#  4606   4766#  4766   4768#  4768   4770#  4770   4776#  4776   4778#  4778   4780#  4780
4782#  4782   4784#  4784   4786#  4786   4788#  4788   4790#  4790   4792#  4792   4794#  4794
4799#  4799   4801#  4801   4803#  4803   4805#  4805   4810#  4810   4812#  4812   4814#  4814
4816#  4816   4818#  4818   4820#  4820   4822#  4822   4824#  4824   4826#  4826   4831#  4831
4833#  4833   4835#  4835   4837#  4837   4839#  4839   4841#  4841   4843#  4843   4845#  4845
4847#  4847   5048#  5048   5050#  5050   5052#  5052   5058#  5058   5060#  5060   5062#  5062
5064#  5064   5066#  5066   5068#  5068   5070#  5070   5072#  5072   5079#  5079   5081#  5081
5083#  5083   5085#  5085   5087#  5087   5089#  5089   5091#  5091   5093#  5093   5098#  5098
5100#  5100   5102#  5102   5104#  5104   5110#  5110   5112#  5112   5114#  5114   5116#  5116
5118#  5118   5124#  5124   5126#  5126   5128#  5128   5130#  5130   5132#  5132   5137#  5137
5139#  5139   5141#  5141   5143#  5143   5145#  5145   5147#  5147   5149#  5149   5151#  5151
5153#  5153   5158#  5158   5160#  5160   5162#  5162   5164#  5164   5166#  5166   5168#  5168
5170#  5170   5172#  5172   5174#  5174   5324#  5324   5326#  5326   5328#  5328   5330#  5330
5332#  5332   5337#  5337   5339#  5339   5341#  5341   5343#  5343   5345#  5345   5347#  5347
5349#  5349   5351#  5351   5353#  5353   5355#  5355   5360#  5360   5362#  5362   5364#  5364
5366#  5366   5368#  5368   5370#  5370   5372#  5372   5374#  5374   5379#  5379   5381#  5381
5383#  5383   5385#  5385   5387#  5387   5389#  5389   5391#  5391   5393#  5393   5398#  5398
5400#  5400   5402#  5402   5404#  5404   5406#  5406   5408#  5408   5413#  5413   5415#  5415
5420#  5420   5422#  5422   5424#  5424   5426#  5426   5428#  5428   5430#  5430   5432#  5432
5434#  5434   5436#  5436   5441#  5441   5443#  5443   5445#  5445   5447#  5447   5449#  5449
5451#  5451   5453#  5453   5455#  5455   5457#  5457   5595#  5595   5597#  5597   5599#  5599
5601#  5601   5606#  5606   5608#  5608   5610#  5610   5612#  5612   5614#  5614   5616#  5616
```

```
5621#  5621   5623#  5623   5625#  5625   5627#  5627   5629#  5629   5631#  5631   5636#  5636        SEQ 1576
5638#  5638   5640#  5640   5642#  5642   5644#  5644   5646#  5646   5648#  5648   5653#  5653
```

```
                5621#  5621    5623#  5623    5625#  5625    5627#  5627    5629#  5629    5631#  5631    5636#  5636      SEQ 1576
                5638#  5638    5640#  5640    5642#  5642    5644#  5644    5646#  5646    5648#  5648    5653#  5653
                5655#  5655    5657#  5657    5659#  5659    5661#  5661    5663#  5663    5665#  5665    5667#  5667
                5669#  5669    5674#  5674    5676#  5676    5678#  5678    5680#  5680    5682#  5682    5684#  5684
                5686#  5686    5688#  5688    5690#  5690    5811#  5811    5813#  5813    5815#  5815    5818#  5818
                5820#  5820    5822#  5822    5941#  5941    5946#  5946    5948#  5948    5950#  5950    5952#  5952
                5954#  5954    5956#  5956    5958#  5958    5960#  5960    5962#  5962    5964#  5964    5966#  5966
                5968#  5968    5970#  5970    5972#  5972    5974#  5974    5976#  5976    5978#  5978    5980#  5980
                5982#  5982    5984#  5984    5986#  5986    5988#  5988    5990#  5990    5992#  5992    5994#  5994
                5996#  5996    5998#  5998    6000#  6000    6002#  6002    6004#  6004    6006#  6006    6008#  6008
                6010#  6010    6012#  6012    6014#  6014    6016#  6016    6018#  6018    6020#  6020    6022#  6022
                6024#  6024    6026#  6026    6028#  6028    6030#  6030    6032#  6032    6034#  6034    6036#  6036
                6038#  6038    6040#  6040    6042#  6042    6044#  6044    6046#  6046    6048#  6048    6050#  6050
                6052#  6052    6054#  6054    6056#  6056    6058#  6058    6060#  6060    6062#  6062    6064#  6064
                6066#  6066    6068#  6068    6070#  6070    6072#  6072    6074#  6074    6076#  6076    6078#  6078
                6080#  6080    6082#  6082    6084#  6084    6086#  6086    6088#  6088    6090#  6090    6092#  6092
                6094#  6094    6096#  6096    6098#  6098    6100#  6100    6102#  6102    6104#  6104    6106#  6106
                6108#  6108    6110#  6110    6112#  6112    6114#  6114    6116#  6116    6118#  6118    6120#  6120
                6122#  6122    6124#  6124    6126#  6126    6128#  6128    6130#  6130    6132#  6132    6134#  6134
                6136#  6136    6138#  6138    6140#  6140    6142#  6142    6144#  6144    6146#  6146    6148#  6148
                6150#  6150    6152#  6152    6154#  6154    6156#  6156    6158#  6158    6160#  6160    6162#  6162
                6164#  6164    6166#  6166    6168#  6168    6173#  6173    6181#  6181    6183#  6183    6185#  6185
                6187#  6187    6189#  6189    6194#  6194    6196#  6196    6198#  6198    6200#  6200    6202#  6202
                6204#  6204    6206#  6206    6208#  6208    6210#  6210    6212#  6212    6214#  6214    6216#  6216
                6218#  6218    6220#  6220    6222#  6222    6224#  6224    6226#  6226    6228#  6228    6230#  6230
                6232#  6232    6234#  6234    6236#  6236    6238#  6238    6240#  6240    6242#  6242    6244#  6244
                6246#  6246    6248#  6248    6250#  6250    6252#  6252    6254#  6254    6256#  6256    6258#  6258
                6263#  6263    6265#  6265    6267#  6267    6269#  6269    6272#  6272    6274#  6274    6276#  6276
                6278#  6278    6281#  6281    6283#  6283    6285#  6285    6287#  6287    6290#  6290    6292#  6292
                6294#  6294    6296#  6296    6299#  6299    6301#  6301    6303#  6303    6305#  6305    6308#  6308
                6310#  6310    6312#  6312    6314#  6314    6317#  6317    6319#  6319    6321#  6321    6323#  6323
                6326#  6326    6328#  6328    6330#  6330    6332#  6332    6335#  6335    6337#  6337    6339#  6339
                6341#  6341    6344#  6344    6346#  6346    6348#  6348    6350#  6350    6353#  6353    6355#  6355
                6357#  6357    6359#  6359    6362#  6362    6364#  6364    6366#  6366    6368#  6368    6371#  6371
                6373#  6373    6375#  6375    6377#  6377    6380#  6380    6382#  6382    6384#  6384    6386#  6386
                6389#  6389    6391#  6391    6393#  6393    6395#  6395    6398#  6398    6400#  6400    6402#  6402
                6404#  6404    6407#  6407    6409#  6409    6411#  6411    6413#  6413    6416#  6416    6418#  6418
                6420#  6420    6422#  6422    6425#  6425    6427#  6427    6429#  6429    6431#  6431    6434#  6434
                6436#  6436    6438#  6438    6440#  6440    6443#  6443    6445#  6445    6447#  6447    6449#  6449
                6452#  6452    6454#  6454    6456#  6456    6458#  6458    6461#  6461    6463#  6463    6465#  6465
                6467#  6467    6470#  6470    6472#  6472    6474#  6474    6476#  6476    6479#  6479    6481#  6481
                6483#  6483    6485#  6485    6488#  6488    6490#  6490    6492#  6492    6494#  6494    6497#  6497
                6499#  64[illegible]  6501#  6501    6503#  6503    6506#  6506    6508#  6508    6510#  6510    6512#  6512
                6515#  6515    6517#  6517    6519#  6519    6521#  6521    6524#  6524    6526#  6526    6528#  6528
                6530#  6530    6533#  6533    6535#  6535    6537#  6537    6539#  6539    6542#  6542    6544#  6544
                6546#  6546    6548#  6548    6551#  6551    6741#  6741    6743#  6743    6745#  6745    6747#  6747
                6753#  6753    6755#  6755    6841#  6841    6843#  6843    6845#  6845    6847#  6847    6852#  6852
                6854#  6854    6859#  6859    6861#  6861    6863#  6863    6865#  6865    6867#  6867    6961#  6961
                6963#  6963    6965#  6965    6967#  6967    6969#  6969
%MR             1213#  1213    1215#  1215    1217#  1217    1220#  1220    1222#  1222    1224#  1224    1227#  1227
                1229#  1229    1231#  1231    1234#  1234    1236#  1236    1238#  1238    1328#  1328    1330#  1330
                1332#  1332    1424#  1424    1426#  1426    1428#  1428    1431#  1431    1433#  1433    1435#  1435
                1438#  1438    1440#  1440    1442#  1442    1445#  1445    1447#  1447    1449#  1449    1452#  1452
```

```
1454#  1454   1456#  1456   1459#  1459   1461#  1461   1463#  1463   1466#  1466   1468#  1468     SEQ 1577
1470#  1470   1473#  1473   1475#  1475   1477#  1477   1564#  1564   1566#  1566   1568#  1568
1571#  1571   1573#  1573   1575#  1575   1578#  1578   1580#  1580   1582#  1582   1749#  1749
1751#  1751   1753#  1753   1758#  1758   1760#  1760   1762#  1762   1764#  1764   1769#  1769
1771#  1771   1773#  1773   1775#  1775   1780#  1780   1782#  1782   1784#  1784   1786#  1786
1788#  1788   1790#  1790   1792#  1792   1797#  1797   1799#  1799   1801#  1801   1803#  1803
1805#  1805   1807#  1807   1809#  1809   1814#  1814   1816#  1816   1818#  1818   1820#  1820
1822#  1822   1824#  1824   1826#  1826   1831#  1831   1833#  1833   1838#  1838   1840#  1840              .MCJ
1842#  1842   1844#  1844   1846#  1846   1848#  1848   1850#  1850   1852#  1852   1854#  1854
1859#  1859   1861#  1861   1863#  1863   1865#  1865   1867#  1867   1869#  1869   1871#  1871
1873#  1873   1875#  1875   2026#  2026   2028#  2028   2030#  2030   2035#  2035   2037#  2037
2039#  2039   2044#  2044   2046#  2046   2048#  2048   2050#  2050   2052#  2052   2054#  2054
2059#  2059   2061#  2061   2063#  2063   2065#  2065   2067#  2067   2069#  2069   2071#  2071
2077#  2077   2079#  2079   2081#  2081   2083#  2083   2085#  2085   2087#  2087   2089#  2089
2091#  2091   2096#  2096   2098#  2098   2100#  2100   2102#  2102   2104#  2104   2106#  2106
2108#  2108   2110#  2110   2112#  2112   2117#  2117   2119#  2119   2121#  2121   2123#  2123
2125#  2125   2127#  2127   2129#  2129   2131#  2131   2133#  2133   2248#  2248   2250#  2250
2252#  2252   2254#  2254   2256#  2256   2258#  2258   2260#  2260   2262#  2262   2264#  2264              .MCJ
2266#  2266   2268#  2268   2270#  2270   2272#  2272   2274#  2274   2276#  2276   2278#  2278
2280#  2280   2282#  2282   2284#  2284   2286#  2286   2288#  2288   2290#  2290   2292#  2292
2294#  2294   2296#  2296   2298#  2298   2300#  2300   2302#  2302   2304#  2304   2306#  2306
2308#  2308   2310#  2310   2312#  2312   2314#  2314   2316#  2316   2318#  2318   2320#  2320
2322#  2322   2327#  2327   2329#  2329   2331#  2331   2336#  2336   2338#  2338   2340#  2340
2342#  2342   2347#  2347   2349#  2349   2351#  2351   2356#  2356   2358#  2358   2360#  2360
2365#  2365   2367#  2367   2369#  2369   2374#  2374   2376#  2376   2378#  2378   2383#  2383
2385#  2385   2387#  2387   2392#  2392   2394#  2394   2396#  2396   2401#  2401   2403#  2403
2405#  2405   2410#  2410   2412#  2412   2414#  2414   2419#  2419   2421#  2421   2423#  2423
2428#  2428   2430#  2430   2432#  2432   2437#  2437   2439#  2439   2441#  2441   2446#  2446              .MCO
2448#  2448   2450#  2450   2455#  2455   2457#  2457   2459#  2459   2464#  2464   2466#  2466
2468#  2468   2473#  2473   2475#  2475   2477#  2477   2482#  2482   2484#  2484   2486#  2486
2491#  2491   2493#  2493   2495#  2495   2500#  2500   2502#  2502   2504#  2504   2509#  2509
2511#  2511   2513#  2513   2518#  2518   2520#  2520   2522#  2522   2527#  2527   2529#  2529
2531#  2531   2536#  2536   2538#  2538   2540#  2540   2545#  2545   2547#  2547   2549#  2549
2554#  2554   2556#  2556   2558#  2558   2563#  2563   2565#  2565   2567#  2567   2572#  2572
2574#  2574   2576#  2576   2581#  2581   2583#  2583   2585#  2585   2590#  2590   2592#  2592
2594#  2594   2599#  2599   2601#  2601   2603#  2603   2608#  2608   2610#  2610   2612#  2612
2617#  2617   2619#  2619   2621#  2621   2626#  2626   2628#  2628   2630#  2630   2635#  2635
2637#  2637   2639#  2639   2644#  2644   2646#  2646   2648#  2648   2653#  2653   2655#  2655
2657#  2657   2659#  2659   2664#  2664   2666#  2666   2671#  2671   2673#  2673   2675#  2675
2680#  2680   2682#  2682   2684#  2684   2686#  2686   2688#  2688   2690#  2690   2692#  2692
2694#  2694   2696#  2696   2701#  2701   2703#  2703   2705#  2705   2707#  2707   2709#  2709
2711#  2711   2713#  2713   2715#  2715   2717#  2717   2848#  2848   2850#  2850   2852#  2852
2857#  2857   2859#  2859   2861#  2861   2863#  2863   2865#  2865   2867#  2867   2872#  2872
2874#  2874   2876#  2876   2878#  2878   2880#  2880   2882#  2882   2887#  2887   2889#  2889
2891#  2891   2893#  2893   2895#  2895   2897#  2897   2899#  2899   2904#  2904   2906#  2906
2908#  2908   2910#  2910   2912#  2912   2914#  2914   2916#  2916   2918#  2918   2920#  2920
2925#  2925   2927#  2927   2929#  2929   2931#  2931   2933#  2933   2935#  2935   2937#  2937
2939#  2939   2941#  2941   3054#  3054   3056#  3056   3058#  3058   3063#  3063   3065#  3065
3067#  3067   3069#  3069   3071#  3071   3073#  3073   3075#  3075   3077#  3077   3082#  3082
3084#  3084   3086#  3086   3088#  3088   3090#  3090   3092#  3092   3094#  3094   3096#  3096
3098#  3098   3103#  3103   3105#  3105   3107#  3107   3109#  3109   3111#  3111   3113#  3113
3115#  3115   3117#  3117   3119#  3119   3225#  3225   3227#  3227   3229#  3229   3231#  3231
3233#  3233   3235#  3235   3237#  3237   3239#  3239   3244#  3244   3246#  3246   3248#  3248              .MCR
```

```
3250#  3250   3252#  3252   3254#  3254   3259#  3259   3261#  3261   3263#  3263   3265#  3265              .MCR
```

565

```
3250#  3250    3252#  3252    3254#  3254    3259#  3259    3261#  3261    3263#  3263    3265#  3265      SEQ 1578
3267#  3267    3269#  3269    3274#  3274    3276#  3276    3278#  3278    3280#  3280    3282#  3282
3284#  3284    3286#  3286    3288#  3288    3290#  3290    3295#  3295    3297#  3297    3299#  3299
3301#  3301    3303#  3303    3305#  3305    3307#  3307    3309#  3309    3311#  3311    3417#  3417
3419#  3419    3421#  3421    3423#  3423    3425#  3425    3427#  3427    3429#  3429    3431#  3431
3436#  3436    3438#  3438    3440#  3440    3442#  3442    3444#  3444    3446#  3446    3451#  3451
3453#  3453    3455#  3455    3457#  3457    3459#  3459    3461#  3461    3466#  3466    3468#  3468
3470#  3470    3472#  3472    3474#  3474    3476#  3476    3478#  3478    3480#  3480    3482#  3482
3487#  3487    3489#  3489    3491#  3491    3493#  3493    3495#  3495    3497#  3497    3499#  3499
3501#  3501    3503#  3503    3609#  3609    3611#  3611    3613#  3613    3615#  3615    3617#  3617
3619#  3619    3621#  3621    3623#  3623    3628#  3628    3630#  3630    3632#  3632    3634#  3634
3636#  3636    3638#  3638    3643#  3643    3645#  3645    3647#  3647    3649#  3649    3651#  3651
3653#  3653    3658#  3658    3660#  3660    3662#  3662    3664#  3664    3666#  3666    3668#  3668
3670#  3670    3672#  3672    3674#  3674    3679#  3679    3681#  3681    3683#  3683    3685#  3685
3687#  3687    3689#  3689    3691#  3691    3693#  3693    3695#  3695    3839#  3839    3841#  3841
3843#  3843    3848#  3848    3850#  3850    3852#  3852    3857#  3857    3859#  3859    3861#  3861
3863#  3863    3865#  3865    3870#  3870    3872#  3872    3874#  3874    3876#  3876    3878#  3878
3880#  3880    3882#  3882    3884#  3884    3886#  3886    3891#  3891    3893#  3893    3895#  3895
3897#  3897    3899#  3899    3901#  3901    3903#  3903    3905#  3905    3907#  3907    4043#  4043
4045#  4045    4047#  4047    4052#  4052    4054#  4054    4056#  4056    4058#  4058    4063#  4063
4065#  4065    4067#  4067    4069#  4069    4074#  4074    4076#  4076    4078#  4078    4080#  4080
4082#  4082    4084#  4084    4086#  4086    4088#  4088    4090#  4090    4095#  4095    4097#  4097
4099#  4099    4101#  4101    4103#  4103    4105#  4105    4107#  4107    4109#  4109    4111#  4111
4235#  4235    4237#  4237    4239#  4239    4244#  4244    4246#  4246    4248#  4248    4253#  4253
4255#  4255    4257#  4257    4262#  4262    4264#  4264    4266#  4266    4271#  4271    4273#  4273
4275#  4275    4277#  4277    4279#  5279    4281#  4281    4283#  4283    4285#  4285    4287#  4287
4292#  4292    4294#  4294    4296#  4296    4298#  4298    4300#  4300    4302#  4302    4304#  4304
4306#  4306    4308#  4308    4472#  4472    4474#  4474    4476#  4476    4482#  4482    4484#  4484
4486#  4486    4488#  4488    4494#  4494    4496#  4496    4498#  4498    4500#  4500    4505#  4505
4507#  4507    4509#  4509    4511#  4511    4513#  4513    4518#  4518    4520#  4520    4522#  4522
4524#  4524    4526#  4526    4531#  4531    4533#  4533    4535#  4535    4540#  4540    4542#  4542
4544#  4544    4546#  4546    4548#  4548    4550#  4550    4552#  4552    4554#  4554    4556#  4556
4558#  4558    4560#  4560    4562#  4562    4564#  4564    4569#  4569    4571#  4571    4573#  4573
4575#  4575    4577#  4577    4579#  4579    4581#  4581    4583#  4583    4585#  4585    4590#  4590
4592#  4592    4594#  4594    4596#  4596    4598#  4598    4600#  4600    4602#  4602    4604#  4604
4606#  4606    4766#  4766    4768#  4768    4770#  4770    4776#  4776    4778#  4778    4780#  4780
4782#  4782    4784#  4784    4786#  4786    4788#  4788    4790#  4790    4792#  4792    4794#  4794
4799#  4799    4801#  4801    4803#  4803    4805#  4805    4810#  4810    4812#  4812    4814#  4814
4816#  4816    4818#  4818    4820#  4820    4822#  4822    4824#  4824    4826#  4826    4831#  4831
4833#  4833    4835#  4835    4837#  4837    4839#  4839    4841#  4841    4843#  4843    4845#  4845
4847#  4847    5048#  5048    5050#  5050    5052#  5052    5058#  5058    5060#  5060    5062#  5062
5064#  5064    5066#  5066    5068#  5068    5070#  5070    5072#  5072    5079#  5079    5081#  5081
5083#  5083    5085#  5085    5087#  5087    5089#  5089    5091#  5091    5093#  5093    5098#  5098
5100#  5100    5102#  5102    5104#  5104    5110#  5110    5112#  5112    5114#  5114    5116#  5116
5118#  5118    5124#  5124    5126#  5126    5128#  5128    5130#  5130    5132#  5132    5137#  5137
5139#  5139    5141#  5141    5143#  5143    5145#  5145    5147#  5147    5149#  5149    5151#  5151
5153#  5153    5158#  5158    5160#  5160    5162#  5162    5164#  5164    5166#  5166    5168#  5168
5170#  5170    5172#  5172    5174#  5174    5324#  5324    5326#  5326    5328#  5328    5330#  5330
5332#  5332    5337#  5337    5339#  5339    5341#  5341    5343#  5343    5345#  5345    5347#  5347
5349#  5349    5351#  5351    5353#  5353    5355#  5355    5360#  5360    5362#  5362    5364#  5364
5366#  5366    5368#  5368    5370#  5370    5372#  5372    5374#  5374    5379#  5379    5381#  5381
5383#  5383    5385#  5385    5387#  5387    5389#  5389    5391#  5391    5393#  5393    5398#  5398
5400#  5400    5402#  5402    5404#  5404    5406#  5406    5408#  5408    5413#  5413    5415#  5415
```

.MCR

.MD

```
5420#  5420    5422#  5422    5424#  5424    5426#  5426    5428#  5428    5430#  5430    5432#  5432
```

```
5420#  5420    5422#  5422    5424#  5424    5426#  5426    5428#  5428    5430#  5430    5432#  5432     SEQ 1579
5434#  5434    5436#  5436    5441#  5441    5443#  5443    5445#  5445    5447#  5447    5449#  5449
5451#  5451    5453#  5453    5455#  5455    5457#  5457    5595#  5595    5597#  5597    5599#  5599
5601#  5601    5606#  5606    5608#  5608    5610#  5610    5612#  5612    5614#  5614    5616#  5616
5621#  5621    5623#  5623    5625#  5625    5627#  5627    5629#  5629    5631#  5631    5636#  5636
5638#  5638    5640#  5640    5642#  5642    5644#  5644    5646#  5646    5648#  5648    5653#  5653
5655#  5655    5657#  5657    5659#  5659    5661#  5661    5663#  5663    5665#  5665    5667#  5667
5669#  5669    5674#  5674    5676#  5676    5678#  5678    5680#  5680    5682#  5682    5684#  5684
5686#  5686    5688#  5688    5690#  5690    5811#  5811    5813#  5813    5815#  5815    5818#  5818
5820#  5820    5822#  5822    5941#  5941    5946#  5946    5948#  5948    5950#  5950    5952#  5952
5954#  5954    5956#  5956    5958#  5958    5960#  5960    5962#  5962    5964#  5964    5966#  5966
5968#  5968    5970#  5970    5972#  5972    5974#  5974    5976#  5976    5978#  5978    5980#  5980
5982#  5982    5984#  5984    5986#  5986    5988#  5988    5990#  5990    5992#  5992    5994#  5994
5996#  5996    5998#  5998    6000#  6000    6002#  6002    6004#  6004    6006#  6006    6008#  6008
6010#  6010    6012#  6012    6014#  6014    6016#  6016    6018#  6018    6020#  6020    6022#  6022
6024#  6024    6026#  6026    6028#  6028    6030#  6030    6032#  6032    6034#  6034    6036#  6036
6038#  6038    6040#  6040    6042#  6042    6044#  6044    6046#  6046    6048#  6048    6050#  6050
6052#  6052    6054#  6054    6056#  6056    6058#  6058    6060#  6060    6062#  6062    6064#  6064
6066#  6066    6068#  6068    6070#  6070    6072#  6072    6074#  6074    6076#  6076    6078#  6078
6080#  6080    6082#  6082    6084#  6084    6086#  6086    6088#  6088    6090#  6090    6092#  6092
6094#  6094    6096#  6096    6098#  6098    6100#  6100    6102#  6102    6104#  6104    6106#  6106
6108#  6108    6110#  6110    6112#  6112    6114#  6114    6116#  6116    6118#  6118    6120#  6120
6122#  6122    6124#  6124    6126#  6126    6128#  6128    6130#  6130    6132#  6132    6134#  6134
6136#  6136    6138#  6138    6140#  6140    6142#  6142    6144#  6144    6146#  6146    6148#  6148
6150#  6150    6152#  6152    6154#  6154    6156#  6156    6158#  6158    6160#  6160    6162#  6162
6164#  6164    6166#  6166    6168#  6168    6173#  6173    6181#  6181    6183#  6183    6185#  6185
6187#  6187    6189#  6189    6194#  6194    6196#  6196    6198#  6198    6200#  6200    6202#  6202
6204#  6204    6206#  6206    6208#  6208    6210#  6210    6212#  6212    6214#  6214    6216#  6216
6218#  6218    6220#  6220    6222#  6222    6224#  6224    6226#  6226    6228#  6228    6230#  6230
6232#  6232    6234#  6234    6236#  6236    6238#  6238    6240#  6240    6242#  6242    6244#  6244
6246#  6246    6248#  6248    6250#  6250    6252#  6252    6254#  6254    6256#  6256    6258#  6258
6263#  6263    6265#  6265    6267#  6267    6269#  6269    6272#  6272    6274#  6274    6276#  6276
6278#  6278    6281#  6281    6283#  6283    6285#  6285    6287#  6287    6290#  6290    6292#  6292
6294#  6294    6296#  6296    6299#  6299    6301#  6301    6303#  6303    6305#  6305    6308#  6308
6310#  6310    6312#  6312    6314#  6314    6317#  6317    6319#  6319    6321#  6321    6323#  6323
6326#  6326    6328#  6328    6330#  6330    6332#  6332    6335#  6335    6337#  6337    6339#  6339
6341#  6341    6344#  6344    6346#  6346    6348#  6348    6350#  6350    6353#  6353    6355#  6355
6357#  6357    6359#  6359    6362#  6362    6364#  6364    6366#  6366    6368#  6368    6371#  6371
6373#  6373    6375#  6375    6377#  6377    6380#  6380    6382#  6382    6384#  6384    6386#  6386
6389#  6389    6391#  6391    6393#  6393    6395#  6395    6398#  6398    6400#  6400    6402#  6402
6404#  6404    6407#  6407    6409#  6409    6411#  6411    6413#  6413    6416#  6416    6418#  6418
6420#  6420    6422#  6422    6425#  6425    6427#  6427    6429#  6429    6431#  6431    6434#  6434
6436#  6436    6438#  6438    6440#  6440    6443#  6443    6445#  6445    6447#  6447    6449#  6449
6452#  6452    6454#  6454    6456#  6456    6458#  6458    6461#  6461    6463#  6463    6465#  6465
6467#  6467    6470#  6470    6472#  6472    6474#  6474    6476#  6476    6479#  6479    6481#  6481
6483#  6483    6485#  6485    6488#  6488    6490#  6490    6492#  6492    6494#  6494    6497#  6497
6499#  6499    6501#  6501    6503#  6503    6506#  6506    6508#  6508    6510#  6510    6512#  6512
6515#  6515    6517#  6517    6519#  6519    6521#  6521    6524#  6524    6526#  6526    6528#  6528
6530#  6530    6533#  6533    6535#  6535    6537#  6537    6539#  6539    6542#  6542    6544#  6544
6546#  6546    6548#  6548    6551#  6551    6741#  6741    6743#  6743    6745#  6745    6747#  6747
6753#  6753    6755#  6755    6841#  6841    6843#  6843    6845#  6845    6847#  6847    6852#  6852
6854#  6854    6859#  6859    6861#  6861    6863#  6863    6865#  6865    6867#  6867    6961#  6961
6963#  6963    6965#  6965    6967#  6967    6969#  6969
```

```
.LA       1844   1846   1852   1865   1867   1873   2102   2104   2110   2123   2125   2131   2686   2688     SEQ 1580
          2694   2707   2709   2715   2910   2912   2918   2931   2933   2939   3088   3090   3096   3109
          3111   3117   3231   3280   3282   3288   3301   3303   3309   3423   3472   3474   3480   3493
          3495   3501   3664   3666   3672   3685   3687   3693   3876   3878   3884   3897   3899   3905
          4080   4082   4088   4101   4103   4109   4277   4279   4285   4298   4300   4306   4575   4577
          4583   4596   4598   4604   4816   4818   4824   4837   4839   4845   5110   5124   5143   5145
          5151   5164   5166   5172   5426   5428   5434   5447   5449   5455   5659   5661   5667   5680
          5682   5688   5948   5950   5952   5954   5956   5958   5960   5962   5964   5966   5968   5970
          5972   5974   5976   5978   5980   5982   5984   5986   5988   5990   5992   5994   5996   5998
          6000   6002   6004   6006   6008   6010   6012   6014   6016   6018   6020   6022   6024   6026
          6028   6030   6032   6034   6036   6038   6040   6042   6044   6046   6048   6050   6052   6054
          6056   6058   6060   6062   6064   6066   6068   6070   6072   6074   6076   6078   6080   6082
          6084   6086   6088   6090   6092   6094   6096   6098   6100   6102   6104   6106   6108   6110
          6112   6114   6116   6118   6120   6122   6124   6126   6128   6130   6132   6134   6136   6138
          6140   6142   6144   6146   6148   6150   6152   6154   6156   6158   6160   6162   6164   6166
          6181   6183   6185   6187   6263   6267   6272   6276   6281   6285   6290   6294   6299   6303
          6308   6312   6317   6321   6326   6330   6335   6339   6344   6348   6353   6357   6362   6366
          6371   6375   6380   6384   6389   6393   6398   6402   6407   6411   6416   6420   6425   6429
          6434   6438   6443   6447   6452   6456   6461   6465   6470   6474   6479   6483   6488   6492
          6497   6501   6506   6510   6515   6519   6524   6528   6533   6537   6542   6546   6969
.LADDR    1213   1328   1424   1564   1749   1758   1769   1780   1797   1814   1831   1838   1859   2026
          2035   2044   2059   2077   2096   2117   2248   2327   2336   2347   2356   2365   2374   2383
          2392   2401   2410   2419   2428   2437   2446   2455   2464   2473   2482   2491   2500   2509
          2518   2527   2536   2545   2554   2563   2572   2581   2590   2599   2608   2617   2626   2635
          2644   2653   2664   2671   2680   2701   2848   2857   2872   2887   2904   2925   3054   3063
          3082   3103   3225   3227   3244   3259   3274   3295   3417   3419   3436   3451   3466   3487
          3609   3611   3628   3643   3658   3679   3839   3848   3857   3870   3891   4043   4052   4063
          4074   4095   4235   4244   4253   4262   4271   4292   4472   4482   4494   4505   4518   4531
          4540   4569   4590   4766   4776   4799   4810   4831   5048   5058   5079   5098   5110   5124
          5137   5158   5324   5337   5360   5379   5398   5413   5420   5441   5595   5606   5621   5636
          5653   5674   5811   5941   5946   6173   6181   6194   6263   6741   6745   6753   6755   6841
          6961
.LAND     1838   1859   2065   2096   2117   2327   2336   2347   2356   2365   2374   2383   2392   2401
          2410   2419   2428   2437   2446   2455   2464   2473   2482   2491   2500   2509   2518   2527
          2536   2545   2554   2563   2572   2581   2590   2599   2608   2617   2626   2635   2644   2653
          2664   2680   2701   2889   2904   2925   3082   3103   3274   3295   3466   3487   3658   3679
          3870   3891   4043   4052   4063   4074   4095   4235   4237   4244   4271   4292   4540   4569
          4590   4776   4810   4831   5137   5158   5402   5420   5441   5606   5621   5653   5674   5811
          5946   6841   6845   6859   6961
.LB       1838   1840   1844   1859   1861   1865   2096   2098   2102   2117   2119   2123   2680   2682
          2686   2701   2703   2707   2904   2906   2910   2925   2927   2931   3082   3084   3088   3103
          3105   3109   3229   3274   3276   3280   3295   3297   3301   3421   3466   3468   3472   3487
          3489   3493   3658   3660   3664   3679   3681   3685   3870   3872   3876   3891   3893   3897
          4074   4076   4080   4095   4097   4101   4271   4273   4277   4292   4294   4298   4569   4571
          4575   4590   4592   4596   4810   4812   4816   4831   4833   4837   5114   5128   5137   5139
          5143   5158   5160   5164   5420   5422   5426   5441   5443   5447   5653   5655   5659   5674
          5676   5680   5946   5948   5950   5952   5954   5956   5958   5960   5962   5964   5966   5968
          5970   5972   5974   5976   5978   5980   5982   5984   5986   5988   5990   5992   5994   5996
          5998   6000   6002   6004   6006   6008   6010   6012   6014   6016   6018   6020   6022   6024
          6026   6028   6030   6032   6034   6036   6038   6040   6042   6044   6046   6048   6050   6052
          6054   6056   6058   6060   6062   6064   6066   6068   6070   6072   6074   6076   6078   6080
          6082   6084   6086   6088   6090   6092   6094   6096   6098   6100   6102   6104   6106   6108
          6110   6112   6114   6116   6118   6120   6122   6124   6126   6128   6130   6132   6134   6136
          6138    6140    6142    6144    6146    6148    6150    6152    6154    6156    6158    6160    6162    6164         SEQ 1581
          6166    6194    6196    6198    6200    6202    6204    6206    6208    6210    6212    6214    6216    6218
          6220    6222    6224    6226    6228    6230    6232    6234    6236    6238    6240    6242    6244    6246
          6248    6250    6252    6254    6256    6265    6267    6269    6274    6276    6278    6283    6285    6287
          6292    6294    6296    6301    6303    6305    6310    6312    6314    6319    6321    6323    6328    6330
          6332    6337    6339    6341    6346    6348    6350    6355    6357    6359    6364    6366    6368    6373
          6375    6377    6382    6384    6386    6391    6393    6395    6400    6402    6404    6409    6411    6413
          6418    6420    6422    6427    6429    6431    6436    6438    6440    6445    6447    6449    6454    6456
          6458    6463    6465    6467    6472    6474    6476    6481    6483    6485    6490    6492    6494    6499
          6501    6503    6508    6510    6512    6517    6519    6521    6526    6528    6530    6535    6537    6539
          6544    6546    6548    6967
.LBAD     1217    1224    1231    1238    1332    1428    1435    1442    1449    1456    1463    1470    1477    1568
          1575    1582    6168    6173
.LCCAF    3237    3252    3259    3263    3267    3451    3455    3459    3636    3643    3647    3651
.LCCBF    3429    3444
.LCCCA    1751    1762    1773    1784    1790    1801    1807    1818    1824    4554    4780    5398    5610    5625
          5640
.LCCCC    2850    2865    2880    3071
.LCCCE    5330    5353    5372    5391
.LCCCP    4768    4784    4803
.LCCEB    3056    3075
.LCCER    2079    3841    3850    3859    3863
.LCCFZ    2329    2340    2664    2673    6267    6276    6285    6294    6303    6312    6321    6330    6339    6348
          6357    6366    6375    6384    6393    6402    6411    6420    6429    6438    6447    6456    6465    6474
          6483    6492    6501    6510    6519    6528    6537    6546
.LCCGC    1848    1869    2028    2037    2046    2052    2061    2069    2083    2106    2127    2690    2711    2859
          2874    2891    2897    2914    2935    3065    3092    3113    3284    3305    3476    3497    3668    3689
          3880    3901    4084    4105    4281    4302    4579    4600    4820    4841    5147    5168    5430    5451
          5663    5684
.LCCIA    4045    4056    4067
.LCCLW    5599    5614    5629    5646
.LCCMB    4237    4246    4255    4264
.LCCMP    4474    4486    4498    4511    4524
.LCCPP    5050    5102    5116    5130
.LCCXA    3621
.LCENA    1751    1762    1773    1784    1790    1801    1807    1818    1824    1848    1869    2028    2037    2046
          2052    2061    2069    2079    2083    2106    2127    2329    2340    2664    2673    2690    2711    2850
          2859    2865    2874    2880    2891    2897    2914    2935    3056    3065    3071    3075    3092    3113
          3237    3252    3259    3263    3267    3284    3305    3429    3444    3451    3455    3459    3476    3497
          3621    3636    3643    3647    3651    3668    3689    3841    3850    3859    3863    3880    3901    4045
          4056    4067    4084    4105    4237    4246    4255    4264    4281    4302    4474    4486    4498    4511
          4524    4554    4579    4600    4768    4780    4784    4803    4820    4841    5050    5102    5116    5130
          5147    5168    5330    5353    5372    5391    5398    5430    5451    5599    5610    5614    5625    5629
          5640    5646    5663    5684    6267    6276    6285    6294    6303    6312    6321    6330    6339    6348
          6357    6366    6375    6384    6393    6402    6411    6420    6429    6438    6447    6456    6465    6474
          6483    6492    6501    6510    6519    6528    6537    6546
.LCJP     1751    1762    1773    1784    1790    1801    1807    1818    1824    1848    1869    2028    2037    2046
          2052    2061    2069    2079    2083    2106    2127    2329    2340    2664    2673    2690    2711    2850
          2859    2865    2874    2880    2891    2897    2914    2935    3056    3065    3071    3075    3092    3113
          3237    3252    3259    3263    3267    3284    3305    3429    3444    3451    3455    3459    3476    3497
          3621    3636    3643    3647    3651    3668    3689    3841    3850    3859    3863    3880    3901    4045
          4056    4067    4084    4105    4237    4246    4255    4264    4281    4302    4474    4486    4498    4511
          4524    4554    4579    4600    4768    4780    4784    4803    4820    4841    5050    5102    5116    5130
```

```
         5147  5168  5330  5353  5372  5391  5398  5430  5451  5599  5610  5614  5625  5629     SEQ 1582
         5640  5646  5663  5684  6267  6276  6285  6294  6303  6312  6321  6330  6339  6348
         6357  6366  6375  6384  6393  6402  6411  6420  6429  6438  6447  6456  6465  6474
         6483  6492  6501  6510  6519  6528  6537  6546
.LCJS    3227  3246  3419  3438  3611  3630  4484  4496  4507  4520  5343  5345  5347  5366
         5385  5387  5948  5950  5954  5956  5960  5962  5966  5968  5972  5974  5978  5980
         5984  5986  5990  5992  5996  5998  6002  6004  6008  6010  6014  6016  6020  6022
         6026  6028  6032  6034  6038  6040  6044  6046  6050  6052  6056  6058  6062  6064
         6068  6070  6074  6076  6080  6082  6086  6088  6092  6094  6098  6100  6104  6106
         6110  6112  6116  6118  6122  6124  6128  6130  6134  6136  6140  6142  6146  6148
         6152  6154  6158  6160  6164  6166  6194  6196  6198  6200  6202  6204  6206  6208
         6210  6212  6214  6216  6218  6220  6222  6224  6226  6228  6230  6232  6234  6236
         6238  6240  6242  6244  6246  6248  6250  6252  6254  6256
.LCONT   1215  1222  1229  1236  1330  1426  1433  1440  1447  1454  1461  1468  1475  1566
         1573  1580  1760  1771  1782  1788  1799  1805  1816  1822  1838  1840  1846  1852
         1859  1861  1867  1873  2050  2065  2067  2085  2089  2096  2098  2104  2110  2117
         2119  2125  2131  2327  2336  2338  2347  2356  2365  2374  2383  2392  2401  2410
         2419  2428  2437  2446  2455  2464  2473  2482  2491  2500  2509  2518  2527  2536
         2545  2554  2563  2572  2581  2590  2599  2608  2617  2626  2635  2644  2653  2680
         2682  2688  2694  2701  2703  2709  2715  2863  2878  2889  2895  2904  2906  2912
         2918  2925  2927  2933  2939  3069  3082  3084  3090  3096  3103  3105  3111  3117
         3229  3231  3233  3235  3248  3250  3274  3276  3282  3288  3295  3297  3303  3309
         3421  3423  3425  3427  3440  3442  3466  3468  3474  3480  3487  3489  3495  3501
         3613  3615  3617  3619  3632  3634  3658  3660  3666  3672  3679  3681  3687  3693
         3870  3872  3878  3884  3891  3893  3899  3905  4054  4065  4074  4076  4082  4088
         4095  4097  4103  4109  4271  4273  4279  4285  4292  4294  4300  4306  4509  4522
         4542  4548  4558  4560  4562  4569  4571  4577  4583  4590  4592  4598  4604  4790
         4801  4810  4812  4818  4824  4831  4833  4839  4845  5100  5112  5114  5126  5128
         5137  5139  5145  5151  5158  5160  5166  5172  5389  5402  5420  5422  5428  5434
         5441  5443  5449  5455  5597  5608  5623  5638  5644  5653  5655  5661  5667  5674
         5676  5682  5688  5813  5815  5820  5822  5946  5952  5958  5964  5970  5976  5982
         5988  5994  6000  6006  6012  6018  6024  6030  6036  6042  6048  6054  6060  6066
         6072  6078  6084  6090  6096  6102  6108  6114  6120  6126  6132  6138  6144  6150
         6156  6162  6263  6265  6272  6274  6281  6283  6290  6292  6299  6301  6308  6310
         6317  6319  6326  6328  6335  6337  6344  6346  6353  6355  6362  6364  6371  6373
         6380  6382  6389  6391  6398  6400  6407  6409  6416  6418  6425  6427  6434  6436
         6443  6445  6452  6454  6461  6463  6470  6472  6479  6481  6488  6490  6497  6499
         6506  6508  6515  6517  6524  6526  6533  6535  6542  6544  6741  6743  6843  6845
         6847  6852  6854  6861  6863  6865  6967
.LCRTN   3261  3265  3269  3453  3457  3461  3645  3649  3653  4564  5408  6258  6551
.LCRY    1840  1861  2098  2119  2338  2349  2358  2367  2376  2385  2394  2403  2412  2421
         2430  2439  2448  2457  2466  2475  2484  2493  2502  2511  2520  2529  2538  2547
         2556  2565  2574  2583  2592  2601  2610  2619  2628  2637  2646  2655  2682  2703
         2906  2927  3084  3105  3276  3297  3468  3489  3660  3681  3872  3893  4076  4097
         4273  4294  4571  4592  4812  4833  5139  5160  5422  5443  5655  5676  5952  6963
.LD      1760  1762  1764  1771  1773  1775  1782  1784  1786  1788  1799  1801  1803  1805
         1816  1818  1820  1822  1838  1840  1844  1846  1848  1850  1852  1854  1859  1861
         1865  1867  1869  1871  1873  1875  2065  2096  2098  2102  2104  2106  2108  2110
         2112  2117  2119  2123  2125  2127  2129  2131  2133  2327  2329  2336  2338  2340
         2347  2349  2356  2358  2365  2367  2374  2376  2383  2385  2392  2394  2401  2403
         2410  2412  2419  2421  2428  2430  2437  2439  2446  2448  2455  2457  2464  2466
         2473  2475  2482  2484  2491  2493  2500  2502  2509  2511  2518  2520  2527  2529
         2536  2538  2545  2547  2554  2556  2563  2565  2572  2574  2581  2583  2590  2592
```

```
      2599  2601  2608  2610  2617  2619  2626  2628  2635  2637  2644  2646  2653  2655     SEQ 1583
      2657  2664  2671  2673  2680  2682  2686  2688  2690  2692  2694  2696  2701  2703
      2707  2709  2711  2713  2715  2717  2859  2861  2863  2874  2876  2878  2889  2891
      2893  2895  2904  2906  2910  2912  2914  2916  2918  2920  2925  2927  2931  2933
      2935  2937  2939  2941  3065  3067  3069  3082  3084  3088  3090  3092  3094  3096
      3098  3103  3105  3109  3111  3113  3115  3117  3119  3229  3231  3233  3235  3250
      3274  3276  3280  3282  3284  3286  3288  3290  3295  3297  3301  3303  3305  3307
      3309  3311  3421  3423  3425  3427  3442  3466  3468  3472  3474  3476  3478  3480
      3482  3487  3489  3493  3495  3497  3499  3501  3503  3613  3615  3617  3619  3634
      3658  3660  3664  3666  3668  3670  3672  3674  3679  3681  3685  3687  3689  3691
      3693  3695  3870  3872  3876  3878  3880  3882  3884  3886  3891  3893  3897  3899
      3901  3903  3905  3907  4043  4052  4063  4074  4076  4080  4082  4084  4086  4088
      4090  4095  4097  4101  4103  4105  4107  4109  4111  4235  4237  4244  4246  4253
      4255  4262  4264  4271  4273  4277  4279  4281  4283  4285  4287  4292  4294  4298
      4300  4302  4304  4306  4308  4540  4542  4544  4546  4548  4550  4552  4554  4556
      4558  4560  4562  4569  4571  4575  4577  4579  4581  4583  4585  4590  4592  4596
      4598  4600  4602  4604  4606  4776  4778  4780  4782  4810  4812  4816  4818  4820
      4822  4824  4826  4831  4833  4837  4839  4841  4843  4845  4847  5058  5060  5062
      5064  5066  5068  5070  5072  5079  5081  5083  5085  5087  5089  5091  5093  5110
      5112  5114  5124  5126  5128  5137  5139  5143  5145  5147  5149  5151  5153  5158
      5160  5164  5166  5168  5170  5172  5174  5343  5345  5347  5366  5385  5387  5398
      5400  5402  5404  5420  5422  5426  5428  5430  5432  5434  5436  5441  5443  5447
      5449  5451  5453  5455  5457  5606  5608  5610  5612  5621  5623  5625  5627  5638
      5640  5642  5653  5655  5659  5661  5663  5665  5667  5669  5674  5676  5680  5682
      5684  5686  5688  5690  5811  5813  5815  5818  5820  5822  5946  5948  5950  5952
      5954  5956  5958  5960  5962  5964  5966  5968  5970  5972  5974  5976  5978  5980
      5982  5984  5986  5988  5990  5992  5994  5996  5998  6000  6002  6004  6006  6008
      6010  6012  6014  6016  6018  6020  6022  6024  6026  6028  6030  6032  6034  6036
      6038  6040  6042  6044  6046  6048  6050  6052  6054  6056  6058  6060  6062  6064
      6066  6068  6070  6072  6074  6076  6078  6080  6082  6084  6086  6088  6090  6092
      6094  6096  6098  6100  6102  6104  6106  6108  6110  6112  6114  6116  6118  6120
      6122  6124  6126  6128  6130  6132  6134  6136  6138  6140  6142  6144  6146  6148
      6150  6152  6154  6156  6158  6160  6162  6164  6166  6181  6183  6185  6187  6194
      6196  6198  6200  6202  6204  6206  6208  6210  6212  6214  6216  6218  6220  6222
      6224  6226  6228  6230  6232  6234  6236  6238  6240  6242  6244  6246  6248  6250
      6252  6254  6256  6263  6265  6267  6269  6272  6274  6276  6278  6281  6283  6285
      6287  6290  6292  6294  6296  6299  6301  6303  6305  6308  6310  6312  6314  6317
      6319  6321  6323  6326  6328  6330  6332  6335  6337  6339  6341  6344  6346  6348
      6350  6353  6355  6357  6359  6362  6364  6366  6368  6371  6373  6375  6377  6380
      6382  6384  6386  6389  6391  6393  6395  6398  6400  6402  6404  6407  6409  6411
      6413  6416  6418  6420  6422  6425  6427  6429  6431  6434  6436  6438  6440  6443
      6445  6447  6449  6452  6454  6456  6458  6461  6463  6465  6467  6470  6472  6474
      6476  6479  6481  6483  6485  6488  6490  6492  6494  6497  6499  6501  6503  6506
      6508  6510  6512  6515  6517  6519  6521  6524  6526  6528  6530  6533  6535  6537
      6539  6542  6544  6546  6548  6741  6743  6753  6755  6841  6843  6845  6847  6852
      6854  6859  6861  6863  6865  6961  6963  6965  6967  6969
.LJ   1213  1217  1220  1224  1227  1231  1234  1238  1328  1332  1424  1428  1431  1435
      1438  1442  1445  1449  1452  1456  1459  1463  1466  1470  1473  1477  1564  1568
      1571  1575  1578  1582  1749  1751  1753  1758  1762  1764  1769  1773  1775  1780
      1784  1786  1790  1792  1797  1801  1803  1807  1809  1814  1818  1820  1824  1826
      1831  1833  1842  1844  1848  1850  1854  1863  1865  1869  1871  1875  2026  2028
      2030  2035  2037  2039  2044  2046  2048  2052  2054  2059  2061  2063  2069  2071
      2077  2079  2081  2083  2087  2091  2100  2102  2106  2108  2112  2121  2123  2127
```

```
        2129   2133   2248   2250   2252   2254   2256   2258   2260   2262   2264   2266   2268   2270      SEQ 1584
        2272   2274   2276   2278   2280   2282   2284   2286   2288   2290   2292   2294   2296   2298
        2300   2302   2304   2306   2308   2310   2312   2314   2316   2318   2320   2322   2329   2331
        2340   2342   2349   2351   2358   2360   2367   2369   2376   2378   2385   2387   2394   2396
        2403   2405   2412   2414   2421   2423   2430   2432   2439   2441   2448   2450   2457   2459
        2466   2468   2475   2477   2484   2486   2493   2495   2502   2504   2511   2513   2520   2522
        2529   2531   2538   2540   2547   2549   2556   2558   2565   2567   2574   2576   2583   2585
        2592   2594   2601   2603   2610   2612   2619   2621   2628   2630   2637   2639   2646   2648
        2655   2657   2659   2664   2666   2671   2673   2675   2684   2686   2690   2692   2696   2705
        2707   2711   2713   2717   2848   2850   2852   2857   2859   2861   2865   2867   2872   2874
        2876   2880   2882   2887   2891   2893   2897   2899   2908   2910   2914   2916   2920   2929
        2931   2935   2937   2941   3054   3056   3058   3063   3065   3067   3071   3073   3075   3077
        3086   3088   3092   3094   3098   3107   3109   3113   3115   3119   3227   3237   3239   3244
        3246   3252   3254   3259   3263   3267   3278   3280   3284   3286   3290   3299   3301   3305
        3307   3311   3419   3429   3431   3436   3438   3444   3446   3451   3455   3459   3470   3472
        3476   3478   3482   3491   3493   3497   3499   3503   3611   3621   3623   3628   3630   3636
        3638   3643   3647   3651   3662   3664   3668   3670   3674   3683   3685   3689   3691   3695
        3839   3841   3843   3848   3850   3852   3857   3859   3861   3863   3865   3874   3876   3880
        3882   3886   3895   3897   3901   3903   3907   4043   4045   4047   4052   4056   4058   4063
        4067   4069   4078   4080   4084   4086   4090   4099   4101   4105   4107   4111   4235   4237
        4239   4244   4246   4248   4253   4255   4257   4262   4264   4266   4275   4277   4281   4283
        4287   4296   4298   4302   4304   4308   4472   4474   4476   4482   4484   4486   4488   4494
        4496   4498   4500   4505   4507   4511   4513   4518   4520   4524   4526   4531   4533   4535
        4540   4544   4546   4550   4552   4554   4556   4573   4575   4579   4581   4585   4594   4596
        4600   4602   4606   4766   4768   4770   4776   4778   4780   4782   4784   4786   4788   4792
        4794   4799   4803   4805   4814   4816   4820   4822   4826   4835   4837   4841   4843   4847
        5048   5050   5052   5058   5060   5062   5064   5066   5068   5070   5072   5079   5081   5083
        5085   5087   5089   5091   5093   5098   5102   5104   5110   5116   5118   5124   5130   5132
        5141   5143   5147   5149   5153   5162   5164   5168   5170   5174   5324   5326   5328   5330
        5332   5337   5339   5341   5343   5345   5347   5349   5351   5353   5355   5360   5362   5364
        5366   5368   5370   5372   5374   5379   5381   5383   5385   5387   5391   5393   5398   5400
        5404   5406   5413   5415   5424   5426   5430   5432   5436   5445   5447   5451   5453   5457
        5595   5599   5601   5606   5610   5612   5614   5616   5621   5625   5627   5629   5631   5636
        5640   5642   5646   5648   5657   5659   5663   5665   5669   5678   5680   5684   5686   5690
        5811   5818   5948   5950   5954   5956   5960   5962   5966   5968   5972   5974   5978   5980
        5984   5986   5990   5992   5996   5998   6002   6004   6008   6010   6014   6016   6020   6022
        6026   6028   6032   6034   6038   6040   6044   6046   6050   6052   6056   6058   6062   6064
        6068   6070   6074   6076   6080   6082   6086   6088   6092   6094   6098   6100   6104   6106
        6110   6112   6116   6118   6122   6124   6128   6130   6134   6136   6140   6142   6146   6148
        6152   6154   6158   6160   6164   6166   6168   6173   6181   6183   6185   6187   6189   6194
        6196   6198   6200   6202   6204   6206   6208   6210   6212   6214   6216   6218   6220   6222
        6224   6226   6228   6230   6232   6234   6236   6238   6240   6242   6244   6246   6248   6250
        6252   6254   6256   6267   6269   6276   6278   6285   6287   6294   6296   6303   6305   6312
        6314   6321   6323   6330   6332   6339   6341   6348   6350   6357   6359   6366   6368   6375
        6377   6384   6386   6393   6395   6402   6404   6411   6413   6420   6422   6429   6431   6438
        6440   6447   6449   6456   6458   6465   6467   6474   6476   6483   6485   6492   6494   6501
        6503   6510   6512   6519   6521   6528   6530   6537   6539   6546   6548   6745   6753   6755
        6841   6859   6867   6961   6963   6965   6969
.LJMAP  1213   1217   1220   1224   1227   1231   1234   1238   1328   1332   1424   1428   1431   1435
        1438   1442   1445   1449   1452   1456   1459   1463   1466   1470   1473   1477   1564   1568
        1571   1575   1578   1582   1749   1753   1758   1764   1769   1775   1780   1785   1792   1797
        1803   1809   1814   1820   1826   1831   1833   1850   1854   1871   1875   2026   2030   2035
        2039   2044   2048   2054   2059   2063   2071   2077   2081   2087   2091   2108   2112   2129
        2133 2248 2250 2252 2254 2256 2258 2260 2262 2264 2266 2268 2270 2272
        2274 2276 2278 2280 2282 2284 2286 2288 2290 2292 2294 2296 2298 2300
        2302 2304 2306 2308 2310 2312 2314 2316 2318 2320 2322 2331 2342 2351
        2360 2369 2378 2387 2396 2405 2414 2423 2432 2441 2450 2459 2468 2477
        2486 2495 2504 2513 2522 2531 2540 2549 2558 2567 2576 2585 2594 2603
        2612 2621 2630 2639 2648 2659 2666 2675 2692 2696 2713 2717 2848 2852
        2857 2861 2867 2872 2876 2882 2887 2893 2899 2916 2920 2937 2941 3054
        3058 3063 3067 3073 3077 3094 3098 3115 3119 3239 3244 3254 3286 3290
        3307 3311 3431 3436 3446 3478 3482 3499 3503 3623 3628 3638 3670 3674
        3691 3695 3839 3843 3848 3852 3857 3861 3865 3882 3886 3903 3907 4043
        4047 4052 4058 4063 4069 4086 4090 4107 4111 4235 4239 4244 4248 4253
        4257 4262 4266 4283 4287 4304 4308 4472 4476 4482 4488 4494 4500 4505
        4513 4518 4526 4531 4533 4535 4540 4556 4581 4585 4602 4606 4766 4770
        4776 4782 4786 4792 4794 4799 4805 4822 4826 4843 4847 5048 5052 5058
        5060 5062 5064 5066 5068 5070 5072 5079 5081 5083 5085 5087 5089 5091
        5093 5098 5104 5110 5118 5124 5132 5149 5153 5170 5174 5324 5332 5337
        5355 5360 5374 5379 5393 5400 5413 5415 5432 5436 5453 5457 5595 5601
        5606 5612 5616 5621 5627 5631 5636 5642 5648 5665 5669 5686 5690 5811
        5818 6168 6173 6181 6183 6185 6187 6189 6269 6278 6287 6296 6305 6314
        6323 6332 6341 6350 6359 6368 6377 6386 6395 6404 6413 6422 6431 6440
        6449 6458 6467 6476 6485 6494 6503 6512 6521 6530 6539 6548 6745 6753
        6755 6841 6859 6867 6961 6969
.LJZ    3225 3417 3609 5941 6747
.LLDCT  1842 1863 2100 2121 2349 2358 2367 2376 2385 2394 2403 2412 2421 2430
        2439 2448 2457 2466 2475 2484 2493 2502 2511 2520 2529 2538 2547 2556
        2565 2574 2583 2592 2601 2610 2619 2628 2637 2646 2655 2684 2705 2908
        2929 3086 3107 3278 3299 3470 3491 3662 3683 3874 3895 4078 4099 4275
        4296 4544 4550 4573 4594 4778 4814 4835 5141 5162 5326 5339 5349 5362
        5368 5381 5404 5424 5445 5657 5678 6963
.LLDLM  5811 5818 6263 6272 6281 6290 6299 6308 6317 6326 6335 6344 6353 6362
        6371 6380 6389 6398 6407 6416 6425 6434 6443 6452 6461 6470 6479 6488
        6497 6506 6515 6524 6533 6542 6743 6843 6847
.LLSAD  6194 6196 6198 6200 6202 6204 6206 6208 6210 6212 6214 6216 6218 6220
        6222 6224 6226 6228 6230 6232 6234 6236 6238 6240 6242 6244 6246 6248
        6250 6252 6254 6256 6841 6845 6859
.LMGC   1229 1236 1330 1433 1440 1447 1454 1461 1468 1475 1566 1573 1580 1758
        1760 1769 1771 1780 1782 1788 1797 1799 1805 1814 1816 1822 1831 1846
        1848 1852 1867 1869 1873 2035 2037 2044 2050 2059 2065 2067 2077 2085
        2089 2104 2106 2110 2125 2127 2131 2688 2690 2694 2709 2711 2715 2859
        2863 2874 2878 2889 2891 2895 2912 2914 2918 2933 2935 2939 3065 3069
        3090 3092 3096 3111 3113 3117 3229 3231 3233 3248 3282 3284 3288 3303
        3305 3309 3421 3423 3425 3440 3474 3476 3480 3495 3497 3501 3613 3632
        3666 3668 3672 3687 3689 3693 3878 3880 3884 3899 3901 3905 4052 4054
        4063 4065 4082 4084 4088 4103 4105 4109 4253 4262 4279 4281 4285 4300
        4302 4306 4509 4522 4531 4533 4540 4542 4548 4558 4560 4562 4564 4577
        4579 4583 4598 4600 4604 4778 4790 4794 4801 4818 4820 4824 4839 4841
        4845 5058 5060 5062 5064 5066 5068 5070 5072 5079 5081 5083 5085 5087
        5089 5091 5093 5100 5110 5112 5114 5124 5126 5128 5145 5147 5151 5166
        5168 5172 5326 5339 5349 5362 5368 5381 5389 5402 5404 5413 5415 5428
        5430 5434 5449 5451 5455 5597 5608 5623 5638 5644 5661 5663 5667 5682
        5684 5688 5811 5813 5815 5818 5820 5822 6181 6183 6185 6187 6194 6196
        6198 6200 6202 6204 6206 6208 6210 6212 6214 6216 6218 6220 6222 6224
        6226 6228 6230 6232 6234 6236 6238 6240 6242 6244 6246 6248 6250 6252
```

```
          6254 6256 6263 6265 6269 6272 6274 6278 6281 6283 6287 6290 6292 6296
          6299 6301 6305 6308 6310 6314 6317 6319 6323 6326 6328 6332 6335 6337
          6341 6344 6346 6350 6353 6355 6359 6362 6364 6368 6371 6373 6377 6380
          6382 6386 6389 6391 6395 6398 6400 6404 6407 6409 6413 6416 6418 6422
          6425 6427 6431 6434 6436 6440 6443 6445 6449 6452 6454 6458 6461 6463
          6467 6470 6472 6476 6479 6481 6485 6488 6490 6494 6497 6499 6503 6506
          6508 6512 6515 6517 6521 6524 6526 6530 6533 6535 6539 6542 6544 6548
          6741 6743 6753 6755 6841 6845 6847 6852 6854 6859 6861 6863 6865
.LOENA    1215 1426 1433 1440 1461 1468 1475 1566 1580 1846 1867 2104 2125 2329
          2340 2673 2688 2709 2889 2912 2933 3090 3111 3231 3282 3303 3423 3474
          3495 3666 3687 3878 3899 4052 4063 4082 4103 4279 4300 4562 4577 4598
          4818 4839 5110 5124 5145 5166 5402 5428 5449 5661 5682 5811 5815 5818
          5822 6181 6183 6185 6187 6263 6272 6281 6290 6299 6308 6317 6326 6335
          6344 6353 6362 6371 6380 6389 6398 6407 6416 6425 6434 6443 6452 6461
          6470 6479 6488 6497 6506 6515 6524 6533 6542 6743 6755 6843 6847 6854
          6865 6969
.LOR      1846 1852 1867 1873 2104 2110 2125 2131 2340 2657 2673 2688 2694 2709
          2715 2863 2878 2895 2912 2918 2933 2939 3069 3090 3096 3111 3117 3229
          3231 3282 3288 3303 3309 3421 3423 3474 3480 3495 3501 3666 3672 3687
          3693 3878 3884 3899 3905 4082 4088 4103 4109 4253 4255 4262 4264 4279
          4285 4300 4306 4560 4562 4577 4583 4598 4604 4818 4824 4839 4845 5058
          5060 5062 5064 5066 5068 5070 5072 5079 5081 5083 5085 5087 5089 5091
          5093 5114 5128 5145 5151 5166 5172 5428 5434 5449 5455 5661 5667 5682
          5688 5813 5815 5820 5822 5948 5954 5958 5960 5964 5966 5970 5972 5976
          5978 5982 5984 5988 5990 5994 5996 6000 6002 6006 6008 6012 6014 6018
          6020 6024 6026 6030 6032 6036 6038 6042 6044 6048 6050 6054 6056 6060
          6062 6066 6068 6072 6074 6078 6080 6084 6086 6090 6092 6096 6098 6102
          6104 6108 6110 6114 6116 6120 6122 6126 6128 6132 6134 6138 6140 6144
          6146 6150 6152 6156 6158 6162 6164 6194 6196 6198 6200 6202 6204 6206
          6208 6210 6212 6214 6216 6218 6220 6222 6224 6226 6228 6230 6232 6234
          6236 6238 6240 6242 6244 6246 6248 6250 6252 6254 6256 6263 6265 6269
          6272 6274 6278 6281 6283 6287 6290 6292 6296 6299 6301 6305 6308 6310
          6314 6317 6319 6323 6326 6328 6332 6335 6337 6341 6344 6346 6350 6353
          6355 6359 6362 6364 6368 6371 6373 6377 6380 6382 6386 6389 6391 6395
          6398 6400 6404 6407 6409 6413 6416 6418 6422 6425 6427 6431 6434 6436
          6440 6443 6445 6449 6452 6454 6458 6461 6463 6467 6470 6472 6476 6479
          6481 6485 6488 6490 6494 6497 6499 6503 6506 6508 6512 6515 6517 6521
          6524 6526 6530 6533 6535 6539 6542 6544 6548 6741 6743 6753 6755 6843
          6852 6854 6965 6967
.LPLUS    1840 1844 1861 1865 2098 2102 2119 2123 2338 2349 2358 2367 2376 2385
          2394 2403 2412 2421 2430 2439 2448 2457 2466 2475 2484 2493 2502 2511
          2520 2529 2538 2547 2556 2565 2574 2583 2592 2601 2610 2619 2628 2637
          2646 2655 2671 2682 2686 2703 2707 2906 2910 2927 2931 3084 3086 3105
          3109 3276 3280 3297 3301 3468 3472 3489 3493 3660 3664 3681 3685 3872
          3876 3893 3897 4076 4080 4097 4101 4273 4277 4294 4298 4571 4575 4592
          4596 4812 4816 4833 4837 5139 5143 5160 5164 5422 5426 5443 5447 5655
          5659 5676 5680 5952 6963 6969
.LRAM     6263 6265 6272 6274 6281 6283 6290 6292 6299 6301 6308 6310 6317 6319
          6326 6328 6335 6337 6344 6346 6353 6355 6362 6364 6371 6373 6380 6382
          6389 6391 6398 6400 6407 6409 6416 6418 6425 6427 6434 6436 6443 6445
          6452 6454 6461 6463 6470 6472 6479 6481 6488 6490 6497 6499 6506 6508
          6515 6517 6524 6526 6533 6535 6542 6544 6852 6863
.LRDLM    1222 1426 1447 1454 1461 1468 1475 1573 1580 5813 5820 6265 6274 6283
```

```
          6292 6301 6310 6319 6328 6337 6346 6355 6364 6373 6382 6391 6400 6409
          6418 6427 6436 6445 6454 6463 6472 6481 6490 6499 6508 6517 6526 6535     SEQ 1587
          6544 6753 6852 6861 6863
.LRPCT    1844 1865 2102 2123 2657 2671 2686 2707 2910 2931 3088 3109 3280 3301
          3472 3493 3664 3685 3876 3897 4080 4101 4277 4298 4546 4552 4575 4596
          4788 4816 4837 5143 5164 5328 5341 5351 5364 5370 5383 5406 5426 5447
          5659 5680 6965
.LSOA     1846 1852 1867 1873 2065 2104 2110 2125 2131 2340 2657 2673 2688 2694
          2709 2715 2863 2878 2895 2912 2918 2933 2939 3069 3090 3096 3111 3117
          3231 3282 3288 3303 3309 3423 3474 3480 3495 3501 3666 3672 3687 3693
          3878 3884 3899 3905 4043 4052 4063 4082 4088 4103 4109 4235 4237 4244
          4255 4264 4279 4285 4300 4306 4562 4577 4583 4598 4604 4818 4824 4839
          4845 5110 5124 5145 5151 5166 5172 5402 5428 5434 5449 5455 5661 5667
          5682 5688 5811 5815 5822 5948 5950 5952 5954 5956 5958 5960 5962 5964
          5966 5968 5970 5972 5974 5976 5978 5980 5982 5984 5986 5988 5990 5992
          5994 5996 5998 6000 6002 6004 6006 6008 6010 6012 6014 6016 6018 6020
          6022 6024 6026 6028 6030 6032 6034 6036 6038 6040 6042 6044 6046 6048
          6050 6052 6054 6056 6058 6060 6062 6064 6066 6068 6070 6072 6074 6076
          6078 6080 6082 6084 6086 6088 6090 6092 6094 6096 6098 6100 6102 6104
          6106 6108 6110 6112 6114 6116 6118 6120 6122 6124 6126 6128 6130 6132
          6134 6136 6138 6140 6142 6144 6146 6148 6150 6152 6154 6156 6158 6160
          6162 6164 6166 6181 6183 6185 6187 6263 6272 6281 6290 6299 6308 6317
          6326 6335 6344 6353 6362 6371 6380 6389 6398 6407 6416 6425 6434 6443
          6452 6461 6470 6479 6488 6497 6506 6515 6524 6533 6542 6743 6755 6841
          6843 6845 6847 6854 6859 6865 6961 6963 6965 6967
.LSOB     1840 1861 2098 2119 2682 2703 2906 2927 3084 3105 3276 3297 3468 3489
          3660 3681 3872 3893 4076 4097 4273 4294 4571 4592 4812 4833 5139 5160
          5422 5443 5655 5676
.LSAB     1844 1865 2102 2123 2329 2338 2349 2358 2367 2376 2385 2394 2403 2412
          2421 2430 2439 2448 2457 2466 2475 2484 2493 2502 2511 2520 2529 2538
          2547 2556 2565 2574 2583 2592 2601 2610 2619 2628 2637 2646 2655 2671
          2686 2707 2910 2931 3088 3109 3280 3301 3472 3493 3664 3685 3876 3897
          4080 4101 4246 4277 4298 4575 4596 4816 4837 5143 5164 5426 5447 5659
          5680 5818 6267 6276 6285 6294 6303 6312 6321 6330 6339 6348 6357 6366
          6375 6384 6393 6402 6411 6420 6429 6438 6447 6456 6465 6474 6483 6492
          6501 6510 6519 6528 6537 6546 6969
.LSDO     1838 1859 2096 2117 2327 2336 2347 2356 2365 2374 2383 2392 2401 2410
          2419 2428 2437 2446 2455 2464 2473 2482 2491 2500 2509 2518 2527 2536
          2545 2554 2563 2572 2581 2590 2599 2608 2617 2626 2635 2644 2653 2664
          2680 2701 2889 2904 2925 3082 3103 3229 3233 3274 3295 3421 3425 3466
          3487 3658 3679 3870 3891 4074 4095 4253 4262 4271 4292 4540 4560 4569
          4590 4776 4810 4831 5058 5060 5062 5064 5066 5068 5070 5072 5079 5081
          5083 5085 5087 5089 5091 5093 5112 5114 5126 5128 5137 5158 5420 5441
          5606 5621 5653 5674 5813 5820 5946 6194 6196 6198 6200 6202 6204 6206
          6208 6210 6212 6214 6216 6218 6220 6222 6224 6226 6228 6230 6232 6234
          6236 6238 6240 6242 6244 6246 6248 6250 6252 6254 6256 6265 6269 6274
          6278 6283 6287 6292 6296 6301 6305 6310 6314 6319 6323 6328 6332 6337
          6341 6346 6350 6355 6359 6364 6368 6373 6377 6382 6386 6391 6395 6400
          6404 6409 6413 6418 6422 6427 6431 6436 6440 6445 6449 6454 6458 6463
          6467 6472 6476 6481 6485 6490 6494 6499 6503 6508 6512 6517 6521 6526
          6530 6535 6539 6544 6548 6741 6753 6852 6861 6863
.LSELC    1760 1771 1782 1788 1799 1805 1816 1831 4531 4542 4548 4558 4564 4778
          4790 4794 5326 5339 5349 5362 5368 5381 5389 5404 5413 5597 5608 5623
```

.RJ

```
          5638
.LSELE    1229 1236 1433 1440 1447 1454 1461 1468 1475 1846 1848 1852 1867 1869     SEQ 1588
```

```
            5638
.LSELE      1229  1236  1433  1440  1447  1454  1461  1468  1475  1846  1848  1852  1867  1869     SEQ 1588
            1873  2035  2037  2044  2050  2059  2065  2067  2077  2085  2089  2104  2106  2110
            2125  2127  2131  2688  2690  2694  2709  2711  2715  2859  2863  2874  2878  2889
            2891  2895  2912  2914  2918  2933  2935  2939  3065  3069  3090  3092  3096  3111
            3113  3117  3282  3284  3288  3303  3305  3309  3474  3476  3480  3495  3497  3501
            3666  3668  3672  3687  3689  3693  3878  3880  3884  3899  3901  3905  4052  4054
            4063  4065  4082  4084  4088  4103  4105  4109  4279  4281  4285  4300  4302  4306
            4562  4577  4579  4583  4598  4600  4604  4818  4820  4824  4839  4841  4845  5145
            5147  5151  5166  5168  5172  5428  5430  5434  5449  5451  5455  5661  5663  5667
            5682  5684  5688  5815  5822  6181  6183  6185  6187  6755  6854  6865
.LSELF      5114  5128  5402
.LSELM      1330  1566  1573  1580  1758  1769  1780  1797  1814  1822  3231  3248  3423  3440
            3632  4509  4522  4533  4540  4560  4801  5100  5110  5124  5415  5644
.LSELP      3233  3425  3613  5112  5126
.LSKCN      4253  4262  6269  6278  6287  6296  6305  6314  6323  6332  6341  6350  6359  6368
            6377  6386  6395  6404  6413  6422  6431  6440  6449  6458  6467  6476  6485  6494
            6503  6512  6521  6530  6539  6548  6741
.LSKMB      6969
.LXNOR      2329  4246  5818  5950  5956  5962  5968  5974  5980  5986  5992  5998  6004  6010
            6016  6022  6028  6034  6040  6046  6052  6058  6064  6070  6076  6082  6088  6094
            6100  6106  6112  6118  6124  6130  6136  6142  6148  6154  6160  6166  6847
.LXOR       6267  6276  6285  6294  6303  6312  6321  6330  6339  6348  6357  6366  6375  6384
            6393  6402  6411  6420  6429  6438  6447  6456  6465  6474  6483  6492  6501  6510
            6519  6528  6537  6546
.MA         1844  1846  1852  1865  1867  1873  2102  2104  2110  2123  2125  2131  2686  2688
            2694  2707  2709  2715  2910  2912  2918  2931  2933  2939  3088  3090  3096  3109
            3111  3117  3231  3280  3282  3288  3301  3303  3309  3423  3472  3474  3480  3493
            3495  3501  3664  3666  3672  3685  3687  3693  3876  3878  3884  3897  3899  3905
            4080  4082  4088  4101  4103  4109  4277  4279  4285  4298  4300  4306  4575  4577
            4583  4596  4598  4604  4816  4818  4824  4837  4839  4845  5110  5124  5143  5145
            5151  5164  5166  5172  5426  5428  5434  5447  5449  5455  5659  5661  5667  5680
            5682  5688  5948  5950  5952  5954  5956  5958  5960  5962  5964  5966  5968  5970
            5972  5974  5976  5978  5980  5982  5984  5986  5988  5990  5992  5994  5996  5998
            6000  6002  6004  6006  6008  6010  6012  6014  6016  6018  6020  6022  6024  6026
            6028  6030  6032  6034  6036  6038  6040  6042  6044  6046  6048  6050  6052  6054
            6056  6058  6060  6062  6064  6066  6068  6070  6072  6074  6076  6078  6080  6082
            6084  6086  6088  6090  6092  6094  6096  6098  6100  6102  6104  6106  6108  6110
            6112  6114  6116  6118  6120  6122  6124  6126  6128  6130  6132  6134  6136  6138
            6140  6142  6144  6146  6148  6150  6152  6154  6156  6158  6160  6162  6164  6166
            6181  6183  6185  6187  6263  6267  6272  6276  6281  6285  6290  6294  6299  6303
            6308  6312  6317  6321  6326  6330  6335  6339  6344  6348  6353  6357  6362  6366
            6371  6375  6380  6384  6389  6393  6398  6402  6407  6411  6416  6420  6425  6429
            6434  6438  6443  6447  6452  6456  6461  6465  6470  6474  6479  6483  6488  6492
            6497  6501  6506  6510  6515  6519  6524  6528  6533  6537  6542  6546  6969
.MAND       1838  1859  2065  2096  2117  2327  2336  2347  2356  2365  2374  2383  2392  2401
            2410  2419  2428  2437  2446  2455  2464  2473  2482  2491  2500  2509  2518  2527
            2536  2545  2554  2563  2572  2581  2590  2599  2608  2617  2626  2635  2644  2653
            2664  2680  2701  2889  2904  2925  3082  3103  3274  3295  3466  3487  3658  3679
            3870  3891  4043  4052  4063  4074  4095  4235  4237  4244  4271  4292  4540  4569
            4590  4776  4810  4831  5137  5158  5402  5420  5441  5606  5621  5653  5674  5811
            5946  6841  6845  6859  6961
.MB         1838  1840  1844  1859  1861  1865  2096  2098  2102  2117  2119  2123  2680  2682
```

```
            2686  2701  2703  2707  2904  2906  2910  2925  2927  2931  3082  3084  3088  3103     SEQ 1589
            3105  3109  3229  3274  3276  3280  3295  3297  3301  3421  3466  3468  3472  3487
```

```
576
             2686  2701  2703  2707  2904  2906  2910  2925  2927  2931  3082  3084  3088  3103      SEQ 1589
             3105  3109  3229  3274  3276  3280  3295  3297  3301  3421  3466  3468  3472  3487
             3489  3493  3658  3660  3664  3679  3681  3685  3870  3872  3876  3891  3893  3897
             4074  4076  4080  4095  4097  4101  4271  4273  4277  4292  4294  4298  4569  4571
             4575  4590  4592  4596  4810  4812  4816  4831  4833  4837  5114  5128  5137  5139
             5143  5158  5160  5164  5420  5422  5426  5441  5443  5447  5653  5655  5659  5674
             5676  5680  5946  5948  5950  5952  5954  5956  5958  5960  5962  5964  5966  5968
             5970  5972  5974  5976  5978  5980  5982  5984  5986  5988  5990  5992  5994  5996
             5998  6000  6002  6004  6006  6008  6010  6012  6014  6016  6018  6020  6022  6024
             6026  6028  6030  6032  6034  6036  6038  6040  6042  6044  6046  6048  6050  6052
             6054  6056  6058  6060  6062  6064  6066  6068  6070  6072  6074  6076  6078  6080
             6082  6084  6086  6088  6090  6092  6094  6096  6098  6100  6102  6104  6106  6108
             6110  6112  6114  6116  6118  6120  6122  6124  6126  6128  6130  6132  6134  6136
             6138  6140  6142  6144  6146  6148  6150  6152  6154  6156  6158  6160  6162  6164
             6166  6194  6196  6198  6200  6202  6204  6206  6208  6210  6212  6214  6216  6218
             6220  6222  6224  6226  6228  6230  6232  6234  6236  6238  6240  6242  6244  6246
             6248  6250  6252  6254  6256  6265  6267  6269  6274  6276  6278  6283  6285  6287
             6292  6294  6296  6301  6303  6305  6310  6312  6314  6319  6321  6323  6328  6330
             6332  6337  6339  6341  6346  6348  6350  6355  6357  6359  6364  6366  6368  6373
             6375  6377  6382  6384  6386  6391  6393  6395  6400  6402  6404  6409  6411  6413
             6418  6420  6422  6427  6429  6431  6436  6438  6440  6445  6447  6449  6454  6456
             6459  6463  6465  6467  6472  6474  6476  6481  6483  6485  6490  6492  6494  6499
             6501  6503  6508  6510  6512  6517  6519  6521  6526  6528  6530  6535  6537  6539
             6544  6546  6548  6967
.MBAD        1217  1224  1231  1238  1332  1428  1435  1442  1449  1456  1463  1470  1477  1568
             1575  1582  6168  6173
.MCCAF       3237  3252  3259  3263  3267  3451  3455  3459  3636  3643  3647  3651
.MCCBF       3429  3444
.MCCCA       1751  1762  1773  1784  1790  1801  1807  1818  1824  4554  4780  5398  5610  5625
             5640
.MCCCC       2850  2865  2880  3071
.MCCCE       5330  5353  5372  5391
.MCCCP       4768  4784  4803
.MCCEB       3056  3075
.MCCER       2079  3841  3850  3859  3863
.MCCFZ       2329  2340  2664  2673  6267  6276  6285  6294  6303  6312  6321  6330  6339  6348
             6357  6366  6375  6384  6393  6402  6411  6420  6429  6438  6447  6456  6465  6474
             6483  6492  6501  6510  6519  6528  6537  6546
.MCCGC       1848  1869  2028  2037  2046  2052  2061  2069  2083  2106  2127  2690  2711  2859
             2874  2891  2897  2914  2935  3065  3092  3113  3284  3305  3476  3497  3668  3689
             3880  3901  4084  4105  4281  4302  4579  4600  4820  4841  5147  5168  5430  5451
             5663  5684
.MCCIA       4045  4056  4067
.MCCLW       5599  5614  5629  5646
.MCCMB       4237  4246  4255  4264
.MCCMP       4474  4486  4498  4511  4524
.MCCPP       5050  5102  5116  5130
.MCCXA       3621
.MCENA       1751  1762  1773  1784  1790  1801  1807  1818  1824  1848  1869  2028  2037  2046
             2052  2061  2069  2079  2083  2106  2127  2329  2340  2664  2673  2690  2711  2850
             2859  2865  2874  2880  2891  2897  2914  2935  3056  3065  3071  3075  3092  3113
             3237  3252  3259  3263  3267  3284  3305  3429  3444  3451  3455  3459  3476  3497
             3621  3636  3643  3647  3651  3668  3689  3841  3850  3859  3863  3880  3901  4045

577
             4056  4067  4084  4105  4237  4246  4255  4264  4281  4302  4474  4486  4498  4511      SEQ 1590
```

577

```
          4056  4067  4084  4105  4237  4246  4255  4264  4281  4302  4474  4486  4498  4511      SEQ 1590
          4524  4554  4579  4600  4768  4780  4784  4803  4820  4841  5050  5102  5116  5130
          5147  5168  5330  5353  5372  5391  5398  5430  5451  5599  5610  5614  5625  5629
          5640  5646  5663  5684  6267  6276  6285  6294  6303  6312  6321  6330  6339  6348
          6357  6366  6375  6384  6393  6402  6411  6420  6429  6438  6447  6456  6465  6474
          6483  6492  6501  6510  6519  6528  6537  6546
.MCJP     1751  1762  1773  1784  1790  1801  1807  1818  1824  1848  1869  2028  2037  2046
          2052  2061  2069  2079  2083  2106  2127  2329  2340  2664  2673  2690  2711  2850
          2859  2865  2874  2880  2891  2897  2914  2935  3056  3065  3071  3075  3092  3113
          3237  3252  3259  3263  3267  3284  3305  3429  3444  3451  3455  3459  3476  3497
          3621  3636  3643  3647  3651  3668  3689  3841  3850  3859  3863  3880  3901  4045
          4056  4067  4084  4105  4237  4246  4255  4264  4281  4302  4474  4486  4498  4511
          4524  4554  4579  4600  4768  4780  4784  4803  4820  4841  5050  5102  5116  5130
          5147  5168  5330  5353  5372  5391  5398  5430  5451  5599  5610  5614  5625  5629
          5640  5646  5663  5684  6267  6276  6285  6294  6303  6312  6321  6330  6339  6348
          6357  6366  6375  6384  6393  6402  6411  6420  6429  6438  6447  6456  6465  6474
          6483  6492  6501  6510  6519  6528  6537  6546
.MCJS     3227  3246  3419  3438  3611  3630  4484  4496  4507  4520  5343  5345  5347  5366
          5385  5387  5948  5950  5954  5956  5960  5962  5966  5968  5972  5974  5978  5980
          5984  5986  5990  5992  5996  5998  6002  6004  6008  6010  6014  6016  6020  6022
          6026  6028  6032  6034  6038  6040  6044  6046  6050  6052  6056  6058  6062  6064
          6068  6070  6074  6076  6080  6082  6086  6088  6092  6094  6098  6100  6104  6106
          6110  6112  6116  6118  6122  6124  6128  6130  6134  6136  6140  6142  6146  6148
          6152  6154  6158  6160  6164  6166  6194  6196  6198  6200  6202  6204  6206  6208
          6210  6212  6214  6216  6218  6220  6222  6224  6226  6228  6230  6232  6234  6236
          6238  6240  6242  6244  6246  6248  6250  6252  6254  6256
.MCONT    1215  1222  1229  1236  1330  1426  1433  1440  1447  1454  1461  1468  1475  1566
          1573  1580  1760  1771  1782  1788  1799  1805  1816  1822  1838  1840  1846  1852
          1859  1861  1867  1873  2050  2065  2067  2085  2089  2096  2098  2104  2110  2117
          2119  2125  2131  2327  2336  2338  2347  2356  2365  2374  2383  2392  2401  2410
          2419  2428  2437  2446  2455  2464  2473  2482  2491  2500  2509  2518  2527  2536
          2545  2554  2563  2572  2581  2590  2599  2608  2617  2626  2635  2644  2653  2680
          2682  2688  2694  2701  2703  2709  2715  2863  2878  2889  2895  2904  2906  2912
          2918  2925  2927  2933  2939  3069  3082  3084  3090  3096  3103  3105  3111  3117
          3229  3231  3233  3235  3248  3250  3274  3276  3282  3288  3295  3297  3303  3309
          3421  3423  3425  3427  3440  3442  3466  3468  3474  3480  3487  3489  3495  3501
          3613  3615  3617  3619  3632  3634  3658  3660  3666  3672  3679  3681  3687  3693
          3870  3872  3878  3884  3891  3893  3899  3905  4054  4065  4074  4076  4082  4088
          4095  4097  4103  4109  4271  4273  4279  4285  4292  4294  4300  4306  4509  4522
          4542  4548  4558  4560  4562  4569  4571  4577  4583  4590  4592  4598  4604  4790
          4801  4810  4812  4818  4824  4831  4833  4839  4845  5100  5112  5114  5126  5128
          5137  5139  5145  5151  5158  5160  5166  5172  5389  5402  5420  5422  5428  5434
          5441  5443  5449  5455  5597  5608  5623  5638  5644  5653  5655  5661  5667  5674
          5676  5682  5688  5813  5815  5820  5822  5946  5952  5958  5964  5970  5976  5982
          5988  5994  6000  6006  6012  6018  6024  6030  6036  6042  6048  6054  6060  6066
          6072  6078  6084  6090  6096  6102  6108  6114  6120  6126  6132  6138  6144  6150
          6156  6162  6263  6265  6272  6274  6281  6283  6290  6292  6299  6301  6308  6310
          6317  6319  6326  6328  6335  6337  6344  6346  6353  6355  6362  6364  6371  6373
          6380  6382  6389  6391  6398  6400  6407  6409  6416  6418  6425  6427  6434  6436
          6443  6445  6452  6454  6461  6463  6470  6472  6479  6481  6488  6490  6497  6499
          6506  6508  6515  6517  6524  6526  6533  6535  6542  6544  6741  6743  6843  6845
          6847  6852  6854  6861  6863  6865  6967
.MCRTN    3261  3265  3269  3453  3457  3461  3645  3649  3653  4564  5408  6258  6551
```

```
578   .MCRY     1840  1861  2098  2119  2338  2349  2358  2367  2376  2385  2394  2403  2412  2421      SEQ 1591
```

```
.MCRY   1840  1861  2098  2119  2338  2349  2358  2367  2376  2385  2394  2403  2412  2421      SEQ 1591
        2430  2439  2448  2457  2466  2475  2484  2493  2502  2511  2520  2529  2538  2547
        2556  2565  2574  2583  2592  2601  2610  2619  2628  2637  2646  2655  2682  2703
        2906  2927  3084  3105  3276  3297  3468  3489  3660  3681  3872  3893  4076  4097
        4273  4294  4571  4592  4812  4833  5139  5160  5422  5443  5655  5676  5952  6963
.MD     1760  1762  1764  1771  1773  1775  1782  1784  1786  1788  1799  1801  1803  1805
        1816  1818  1820  1822  1838  1840  1844  1846  1848  1850  1852  1854  1859  1861
        1865  1867  1869  1871  1873  1875  2065  2096  2098  2102  2104  2106  2108  2110
        2112  2117  2119  2123  2125  2127  2129  2131  2133  2327  2329  2336  2338  2340
        2347  2349  2356  2358  2365  2367  2374  2376  2383  2385  2392  2394  2401  2403
        2410  2412  2419  2421  2428  2430  2437  2439  2446  2448  2455  2457  2464  2466
        2473  2475  2482  2484  2491  2493  2500  2502  2509  2511  2518  2520  2527  2529
        2536  2538  2545  2547  2554  2556  2563  2565  2572  2574  2581  2583  2590  2592
        2599  2601  2608  2610  2617  2619  2626  2628  2635  2637  2644  2646  2653  2655
        2657  2664  2671  2673  2680  2682  2686  2688  2690  2692  2694  2696  2701  2703
        2707  2709  2711  2713  2715  2717  2859  2861  2863  2874  2876  2878  2889  2891
        2893  2895  2904  2906  2910  2912  2914  2916  2918  2920  2925  2927  2931  2933
        2935  2937  2939  2941  3065  3067  3069  3082  3084  3088  3090  3092  3094  3096
        3098  3103  3105  3109  3111  3113  3115  3117  3119  3229  3231  3233  3235  3250
        3274  3276  3280  3282  3284  3286  3288  3290  3295  3297  3301  3303  3305  3307
        3309  3311  3421  3423  3425  3427  3442  3466  3468  3472  3474  3476  3478  3480
        3482  3487  3489  3493  3495  3497  3499  3501  3503  3613  3615  3617  3619  3634
        3658  3660  3664  3666  3668  3670  3672  3674  3679  3681  3685  3687  3689  3691
        3693  3695  3870  3872  3876  3878  3880  3882  3884  3886  3891  3893  3897  3899
        3901  3903  3905  3907  4043  4052  4063  4074  4076  4080  4082  4084  4086  4088
        4090  4095  4097  4101  4103  4105  4107  4109  4111  4235  4237  4244  4246  4253
        4255  4262  4264  4271  4273  4277  4279  4281  4283  4285  4287  4292  4294  4298
        4300  4302  4304  4306  4308  4540  4542  4544  4546  4548  4550  4552  4554  4556
        4558  4560  4562  4569  4571  4575  4577  4579  4581  4583  4585  4590  4592  4596
        4598  4600  4602  4604  4606  4776  4778  4780  4782  4810  4812  4816  4818  4820
        4822  4824  4826  4831  4833  4837  4839  4841  4843  4845  4847  5058  5060  5062
        5064  5066  5068  5070  5072  5079  5081  5083  5085  5087  5089  5091  5093  5110
        5112  5114  5124  5126  5128  5137  5139  5143  5145  5147  5149  5151  5153  5158
        5160  5164  5166  5168  5170  5172  5174  5343  5345  5347  5366  5385  5387  5398
        5400  5402  5404  5420  5422  5426  5428  5430  5432  5434  5436  5441  5443  5447
        5449  5451  5453  5455  5457  5606  5608  5610  5612  5621  5623  5625  5627  5638
        5640  5642  5653  5655  5659  5661  5663  5665  5667  5669  5674  5676  5680  5682
        5684  5686  5688  5690  5811  5813  5815  5818  5820  5822  5946  5948  5950  5952
        5954  5956  5958  5960  5962  5964  5966  5968  5970  5972  5974  5976  5978  5980
        5982  5984  5986  5988  5990  5992  5994  5996  5998  6000  6002  6004  6006  6008
        6010  6012  6014  6016  6018  6020  6022  6024  6026  6028  6030  6032  6034  6036
        6038  6040  6042  6044  6046  6048  6050  6052  6054  6056  6058  6060  6062  6064
        6066  6068  6070  6072  6074  6076  6078  6080  6082  6084  6086  6088  6090  6092
        6094  6096  6098  6100  6102  6104  6106  6108  6110  6112  6114  6116  6118  6120
        6122  6124  6126  6128  6130  6132  6134  6136  6138  6140  6142  6144  6146  6148
        6150  6152  6154  6156  6158  6160  6162  6164  6166  6181  6183  6185  6187  6194
        6196  6198  6200  6202  6204  6206  6208  6210  6212  6214  6216  6218  6220  6222
        6224  6226  6228  6230  6232  6234  6236  6238  6240  6242  6244  6246  6248  6250
        6252  6254  6256  6263  6265  6267  6269  6272  6274  6276  6278  6281  6283  6285
        6287  6290  6292  6294  6296  6299  6301  6303  6305  6308  6310  6312  6314  6317
        6319  6321  6323  6326  6328  6330  6332  6335  6337  6339  6341  6344  6346  6348
        6350  6353  6355  6357  6359  6362  6364  6366  6368  6371  6373  6375  6377  6380
        6382  6384  6386  6389  6391  6393  6395  6398  6400  6402  6404  6407  6409  6411
```

```
         6413  6416  6418  6420  6422  6425  6427  6429  6431  6434  6436  6438  6440  6443      SEQ 1592
         6445  6447  6449  6452  6454  6456  6458  6461  6463  6465  6467  6470  6472  6474
         6476  6479  6481  6483  6485  6488  6490  6492  6494  6497  6499  6501  6503  6506
         6508  6510  6512  6515  6517  6519  6521  6524  6526  6528  6530  6533  6535  6537
         6539  6542  6544  6546  6548  6741  6743  6753  6755  6841  6843  6845  6847  6852
         6854  6859  6861  6863  6865  6961  6963  6965  6967  6969
.MJ      1213  1217  1220  1224  1227  1231  1234  1238  1328  1332  1424  1428  1431  1435
         1438  1442  1445  1449  1452  1456  1459  1463  1466  1470  1473  1477  1564  1568
         1571  1575  1578  1582  1749  1751  1753  1758  1762  1764  1769  1773  1775  1780
         1784  1786  1790  1792  1797  1801  1803  1807  1809  1814  1818  1820  1824  1826
         1831  1833  1842  1844  1848  1850  1854  1863  1865  1869  1871  1875  2026  2028
         2030  2035  2037  2039  2044  2046  2048  2052  2054  2059  2061  2063  2069  2071
         2077  2079  2081  2083  2087  2091  2100  2102  2106  2108  2112  2121  2123  2127
         2129  2133  2248  2250  2252  2254  2256  2258  2260  2262  2264  2266  2268  2270
         2272  2274  2276  2278  2280  2282  2284  2286  2288  2290  2292  2294  2296  2298
         2300  2302  2304  2306  2308  2310  2312  2314  2316  2318  2320  2322  2329  2331
         2340  2342  2349  2351  2358  2360  2367  2369  2376  2378  2385  2387  2394  2396
         2403  2405  2412  2414  2421  2423  2430  2432  2439  2441  2448  2450  2457  2459
         2466  2468  2475  2477  2484  2486  2493  2495  2502  2504  2511  2513  2520  2522
         2529  2531  2538  2540  2547  2549  2556  2558  2565  2567  2574  2576  2583  2585
         2592  2594  2601  2603  2610  2612  2619  2621  2628  2630  2637  2639  2646  2648
         2655  2657  2659  2664  2666  2671  2673  2675  2684  2686  2690  2692  2696  2705
         2707  2711  2713  2717  2848  2850  2852  2857  2859  2861  2865  2867  2872  2874
         2876  2880  2882  2887  2891  2893  2897  2899  2908  2910  2914  2916  2920  2929
         2931  2935  2937  2941  3054  3056  3058  3063  3065  3067  3071  3073  3075  3077
         3086  3088  3092  3094  3098  3107  3109  3113  3115  3119  3227  3237  3239  3244
         3246  3252  3254  3259  3263  3267  3278  3280  3284  3286  3290  3299  3301  3305
         3307  3311  3419  3429  3431  3436  3438  3444  3446  3451  3455  3459  3470  3472
         3476  3478  3482  3491  3493  3497  3499  3503  3611  3621  3623  3628  3630  3636
         3638  3643  3647  3651  3662  3664  3668  3670  3674  3683  3685  3689  3691  3695
         3839  3841  3843  3848  3850  3852  3857  3859  3861  3863  3865  3874  3876  3880
         3882  3886  3895  3897  3901  3903  3907  4043  4045  4047  4052  4056  4058  4063
         4067  4069  4078  4080  4084  4086  4090  4099  4101  4105  4107  4111  4235  4237
         4239  4244  4246  4248  4253  4255  4257  4262  4264  4266  4275  4277  4281  4283
         4287  4296  4298  4302  4304  4308  4472  4474  4476  4482  4484  4486  4488  4494
         4496  4498  4500  4505  4507  4511  4513  4518  4520  4524  4526  4531  4533  4535
         4540  4544  4546  4550  4552  4554  4556  4573  4575  4579  4581  4585  4594  4596
         4600  4602  4606  4766  4768  4770  4776  4778  4780  4782  4784  4786  4788  4792
         4794  4799  4803  4805  4814  4816  4820  4822  4826  4835  4837  4841  4843  4847
         5048  5050  5052  5058  5060  5062  5064  5066  5068  5070  5072  5079  5081  5083
         5085  5087  5089  5091  5093  5098  5102  5104  5110  5116  5118  5124  5130  5132
         5141  5143  5147  5149  5153  5162  5164  5168  5170  5174  5324  5326  5328  5330
         5332  5337  5339  5341  5343  5345  5347  5349  5351  5353  5355  5360  5362  5364
         5366  5368  5370  5372  5374  5379  5381  5383  5385  5387  5391  5393  5398  5400
         5404  5406  5413  5415  5424  5426  5430  5432  5436  5445  5447  5451  5453  5457
         5595  5599  5601  5606  5610  5612  5614  5616  5621  5625  5627  5629  5631  5636
         5640  5642  5646  5648  5657  5659  5663  5665  5669  5678  5680  5684  5686  5690
         5811  5818  5948  5950  5954  5956  5960  5962  5966  5968  5972  5974  5978  5980
         5984  5986  5990  5992  5996  5998  6002  6004  6008  6010  6014  6016  6020  6022
         6026  6028  6032  6034  6038  6040  6044  6046  6050  6052  6056  6058  6062  6064
         6068  6070  6074  6076  6080  6082  6086  6088  6092  6094  6098  6100  6104  6106
         6110  6112  6116  6118  6122  6124  6128  6130  6134  6136  6140  6142  6146  6148
         6152  6154  6158  6160  6164  6166  6168  6173  6181  6183  6185  6187  6189  6194
```

1580

```
          6196  6198  6200  6202  6204  6206  6208  6210  6212  6214  6216  6218  6220  6222     SEQ 1593
          6224  6226  6228  6230  6232  6234  6236  6238  6240  6242  6244  6246  6248  6250
          6252  6254  6256  6267  6269  6276  6278  6285  6287  6294  6296  6303  6305  6312
          6314  6321  6323  6330  6332  6339  6341  6348  6350  6357  6359  6366  6368  6375
          6377  6384  6386  6393  6395  6402  6404  6411  6413  6420  6422  6429  6431  6438
          6440  6447  6449  6456  6458  6465  6467  6474  6476  6483  6485  6492  6494  6501
          6503  6510  6512  6519  6521  6528  6530  6537  6539  6546  6548  6745  6753  6755
          6841  6859  6867  6961  6963  6965  6969
.MJMAP    1213  1217  1220  1224  1227  1231  1234  1238  1328  1332  1424  1428  1431  1435
          1438  1442  1445  1449  1452  1456  1459  1463  1466  1470  1473  1477  1564  1568
          1571  1575  1578  1582  1749  1753  1758  1764  1769  1775  1780  1786  1792  1797
          1803  1809  1814  1820  1826  1831  1833  1850  1854  1871  1875  2026  2030  2035
          2039  2044  2048  2054  2059  2063  2071  2077  2081  2087  2091  2108  2112  2129
          2133  2248  2250  2252  2254  2256  2258  2260  2262  2264  2266  2268  2270  2272
          2274  2276  2278  2280  2282  2284  2286  2288  2290  2292  2294  2296  2298  2300
          2302  2304  2306  2308  2310  2312  2314  2316  2318  2320  2322  2331  2342  2351
          2360  2369  2378  2387  2396  2405  2414  2423  2432  2441  2450  2459  2468  2477
          2486  2495  2504  2513  2522  2531  2540  2549  2558  2567  2576  2585  2594  2603
          2612  2621  2630  2639  2648  2659  2666  2675  2692  2696  2713  2717  2848  2852
          2857  2861  2867  2872  2876  2882  2887  2893  2899  2916  2920  2937  2941  3054
          3058  3063  3067  3073  3077  3094  3098  3115  3119  3239  3244  3254  3286  3290
          3307  3311  3431  3436  3446  3478  3482  3499  3503  3623  3628  3638  3670  3674
          3691  3695  3839  3843  3848  3852  3857  3861  3865  3882  3886  3903  3907  4043
          4047  4052  4058  4063  4069  4086  4090  4107  4111  4235  4239  4244  4248  4253
          4257  4262  4266  4283  4287  4304  4308  4472  4476  4482  4488  4494  4500  4505
          4513  4518  4526  4531  4533  4535  4540  4556  4581  4585  4602  4606  4766  4770
          4776  4782  4786  4792  4794  4799  4805  4822  4826  4843  4847  5048  5052  5058
          5060  5062  5064  5066  5068  5070  5072  5079  5081  5083  5085  5087  5089  5091
          5093  5098  5104  5110  5118  5124  5132  5149  5153  5170  5174  5324  5332  5337
          5355  5360  5374  5379  5393  5400  5413  5415  5432  5436  5453  5457  5595  5601
          5606  5612  5616  5621  5627  5631  5636  5642  5648  5665  5669  5686  5690  5811
          5818  6168  6173  6181  6183  6185  6187  6189  6269  6278  6287  6296  6305  6314
          6323  6332  6341  6350  6359  6368  6377  6386  6395  6404  6413  6422  6431  6440
          6449  6458  6467  6476  6485  6494  6503  6512  6521  6530  6539  6548  6745  6753
          6755  6841  6859  6867  6961  6969
.MJZ      3225  3417  3609  5941  6747
.MLDCT    1842  1863  2100  2121  2349  2358  2367  2376  2385  2394  2403  2412  2421  2430
          2439  2448  2457  2466  2475  2484  2493  2502  2511  2520  2529  2538  2547  2556
          2565  2574  2583  2592  2601  2610  2619  2628  2637  2646  2655  2684  2705  2908
          2929  3086  3107  3278  3299  3470  3491  3662  3683  3874  3895  4078  4099  4275
          4296  4544  4550  4573  4594  4778  4814  4835  5141  5162  5326  5339  5349  5362
          5368  5381  5404  5424  5445  5657  5678  6963
.MLDLM    5811  5818  6263  6272  6281  6290  6299  6308  6317  6326  6335  6344  6353  6362
          6371  6380  6389  6398  6407  6416  6425  6434  6443  6452  6461  6470  6479  6488
          6497  6506  6515  6524  6533  6542  6743  6843  6847
.MLSAD    6194  6196  6198  6200  6202  6204  6206  6208  6210  6212  6214  6216  6218  6220
          6222  6224  6226  6228  6230  6232  6234  6236  6238  6240  6242  6244  6246  6248
          6250  6252  6254  6256  6841  6845  6859
.MMGC     1229  1236  1330  1433  1440  1447  1454  1461  1468  1475  1566  1573  1580  1758
          1760  1769  1771  1780  1782  1788  1797  1799  1805  1814  1816  1822  1831  1846
          1848  1852  1867  1869  1873  2035  2037  2044  2050  2059  2065  2067  2077  2085
          2089  2104  2106  2110  2125  2127  2131  2688  2690  2694  2709  2711  2715  2859
          2863  2874  2878  2889  2891  2895  2912  2914  2918  2933  2935  2939  3065  3069
```

CON

1581

```
          3090  3092  3096  3111  3113  3117  3229  3231  3233  3248  3282  3284  3288  3303     SEQ 1594
```

```
1581                                                                                                   SEQ 1594
         3090  3092  3096  3111  3113  3117  3229  3231  3233  3248  3282  3284  3288  3303
         3305  3309  3421  3423  3425  3440  3474  3476  3480  3495  3497  3501  3613  3632
         3666  3668  3672  3687  3689  3693  3878  3880  3884  3899  3901  3905  4052  4054
         4063  4065  4082  4084  4088  4103  4105  4109  4253  4262  4279  4281  4285  4300
         4302  4306  4509  4522  4531  4533  4540  4542  4548  4558  4560  4562  4564  4577
         4579  4583  4598  4600  4604  4778  4790  4794  4801  4818  4820  4824  4839  4841
         4845  5058  5060  5062  5064  5066  5068  5070  5072  5079  5081  5083  5085  5087
         5089  5091  5093  5100  5110  5112  5114  5124  5126  5128  5145  5147  5151  5166
         5168  5172  5326  5339  5349  5362  5368  5381  5389  5402  5404  5413  5415  5428
         5430  5434  5449  5451  5455  5597  5608  5623  5638  5644  5661  5663  5667  5682
         5684  5688  5811  5813  5815  5818  5820  5822  6181  6183  6185  6187  6194  6196
         6198  6200  6202  6204  6206  6208  6210  6212  6214  6216  6218  6220  6222  6224
         6226  6228  6230  6232  6234  6236  6238  6240  6242  6244  6246  6248  6250  6252
         6254  6256  6263  6265  6269  6272  6274  6278  6281  6283  6287  6290  6292  6296
         6299  6301  6305  6308  6310  6314  6317  6319  6323  6326  6328  6332  6335  6337
         6341  6344  6346  6350  6353  6355  6359  6362  6364  6368  6371  6373  6377  6380
         6382  6386  6389  6391  6395  6398  6400  6404  6407  6409  6413  6416  6418  6422
         6425  6427  6431  6434  6436  6440  6443  6445  6449  6452  6454  6458  6461  6463
         6467  6470  6472  6476  6479  6481  6485  6488  6490  6494  6497  6499  6503  6506
         6508  6512  6515  6517  6521  6524  6526  6530  6533  6535  6539  6542  6544  6548
         6741  6743  6753  6755  6841  6845  6847  6852  6854  6859  6861  6863  6865
.MOENA   1215  1426  1433  1440  1461  1468  1475  1566  1580  1846  1867  2104  2125  2329
         2340  2673  2688  2709  2889  2912  2933  3090  3111  3231  3282  3303  3423  3474
         3495  3666  3687  3878  3899  4052  4063  4082  4103  4279  4300  4562  4577  4598
         4818  4839  5110  5124  5145  5166  5402  5428  5449  5661  5682  5811  5815  5818
         5822  6181  6183  6185  6187  6263  6272  6281  6290  6299  6308  6317  6326  6335
         6344  6353  6362  6371  6380  6389  6398  6407  6416  6425  6434  6443  6452  6461
         6470  6479  6488  6497  6506  6515  6524  6533  6542  6743  6755  6843  6847  6854
         6865  6969
.MOR     1846  1852  1867  1873  2104  2110  2125  2131  2340  2657  2673  2688  2694  2709
         2715  2863  2878  2895  2912  2918  2933  2939  3069  3090  3096  3111  3117  3229
         3231  3282  3288  3303  3309  3421  3423  3474  3480  3495  3501  3666  3672  3687
         3693  3878  3884  3899  3905  4082  4088  4103  4109  4253  4255  4262  4264  4279
         4285  4300  4306  4560  4562  4577  4583  4598  4604  4818  4824  4839  4845  5058
         5060  5062  5064  5066  5068  5070  5072  5079  5081  5083  5085  5087  5089  5091
         5093  5114  5128  5145  5151  5166  5172  5428  5434  5449  5455  5661  5667  5682
         5688  5813  5815  5820  5822  5948  5954  5958  5960  5964  5966  5970  5972  5976
         5978  5982  5984  5988  5990  5994  5996  6000  6002  6006  6008  6012  6014  6018
         6020  6024  6026  6030  6032  6036  6038  6042  6044  6048  6050  6054  6056  6060
         6062  6066  6068  6072  6074  6078  6080  6084  6086  6090  6092  6096  6098  6102
         6104  6108  6110  6114  6116  6120  6122  6126  6128  6132  6134  6138  6140  6144
         6146  6150  6152  6156  6158  6162  6164  6194  6196  6198  6200  6202  6204  6206
         6208  6210  6212  6214  6216  6218  6220  6222  6224  6226  6228  6230  6232  6234
         6236  6238  6240  6242  6244  6246  6248  6250  6252  6254  6256  6263  6265  6269
         6272  6274  6278  6281  6283  6287  6290  6292  6296  6299  6301  6305  6308  6310
         6314  6317  6319  6323  6326  6328  6332  6335  6337  6341  6344  6346  6350  6353
         6355  6359  6362  6364  6368  6371  6373  6377  6380  6382  6386  6389  6391  6395
         6398  6400  6404  6407  6409  6413  6416  6418  6422  6425  6427  6431  6434  6436
         6440  6443  6445  6449  6452  6454  6458  6461  6463  6467  6470  6472  6476  6479
         6481  6485  6488  6490  6494  6497  6499  6503  6506  6508  6512  6515  6517  6521
         6524  6526  6530  6533  6535  6539  6542  6544  6548  6741  6743  6753  6755  6843
         6852  6854  6965  6967
.MPLUS   1840  1844  1861  1865  2098  2102  2119  2123  2338  2349  2358  2367  2376  2385


         2394  2403  2412  2421  2430  2439  2448  2457  2466  2475  2484  2493  2502  2511   SEQ 1595
```

```
1582                                                                                              SEQ 1595

                2394  2403  2412  2421  2430  2439  2448  2457  2466  2475  2484  2493  2502  2511
                2520  2529  2538  2547  2556  2565  2574  2583  2592  2601  2610  2619  2628  2637
                2646  2655  2671  2682  2686  2703  2707  2906  2910  2927  2931  3084  3088  3105
                3109  3276  3280  3297  3301  3468  3472  3489  3493  3660  3664  3681  3685  3872
                3876  3893  3897  4076  4080  4097  4101  4273  4277  4294  4298  4571  4575  4592
                4596  4812  4816  4833  4837  5139  5143  5160  5164  5422  5426  5443  5447  5655
                5659  5676  5680  5952  6963  6969
.MRAM           6263  6265  6272  6274  6281  6283  6290  6292  6299  6301  6308  6310  6317  6319
                6326  6328  6335  6337  6344  6346  6353  6355  6362  6364  6371  6373  6380  6382
                6389  6391  6398  6400  6407  6409  6416  6418  6425  6427  6434  6436  6443  6445
                6452  6454  6461  6463  6470  6472  6479  6481  6488  6490  6497  6499  6506  6508
                6515  6517  6524  6526  6533  6535  6542  6544  6852  6863
.MRDLM          1222  1426  1447  1454  1461  1468  1475  1573  1580  5813  5820  6265  6274  6283
                6292  6301  6310  6319  6328  6337  6346  6355  6364  6373  6382  6391  6400  6409
                6418  6427  6436  6445  6454  6463  6472  6481  6490  6499  6508  6517  6526  6535
                6544  6753  6852  6861  6863
.MRPCT          1844  1865  2102  2123  2657  2671  2686  2707  2910  2931  3088  3109  3280  3301
                3472  3493  3664  3685  3876  3897  4080  4101  4277  4298  4546  4552  4575  4596
                4788  4816  4837  5143  5164  5328  5341  5351  5364  5370  5383  5406  5426  5447
                5659  5680  6965
.MSOA           1846  1852  1867  1873  2065  2104  2110  2125  2131  2340  2657  2673  2688  2694
                2709  2715  2863  2878  2895  2912  2918  2933  2939  3069  3090  3096  3111  3117
                3231  3282  3288  3303  3309  3423  3474  3480  3495  3501  3666  3672  3687  3693
                3878  3884  3899  3905  4043  4052  4063  4082  4088  4103  4109  4235  4237  4244
                4255  4264  4279  4285  4300  4306  4562  4577  4583  4598  4604  4818  4824  4839
                4845  5110  5124  5145  5151  5166  5172  5402  5428  5434  5449  5455  5661  5667
                5682  5688  5811  5815  5822  5948  5950  5952  5954  5956  5958  5960  5962  5964
                5966  5968  5970  5972  5974  5976  5978  5980  5982  5984  5986  5988  5990  5992
                5994  5996  5998  6000  6002  6004  6006  6008  6010  6012  6014  6016  6018  6020
                6022  6024  6026  6028  6030  6032  6034  6036  6038  6040  6042  6044  6046  6048
                6050  6052  6054  6056  6058  6060  6062  6064  6066  6068  6070  6072  6074  6076
                6078  6080  6082  6084  6086  6088  6090  6092  6094  6096  6098  6100  6102  6104
                6106  6108  6110  6112  6114  6116  6118  6120  6122  6124  6126  6128  6130  6132
                6134  6136  6138  6140  6142  6144  6146  6148  6150  6152  6154  6156  6158  6160
                6162  6164  6166  6181  6183  6185  6187  6263  6272  6281  6290  6299  6308  6317
                6326  6335  6344  6353  6362  6371  6380  6389  6398  6407  6416  6425  6434  6443
                6452  6461  6470  6479  6488  6497  6506  6515  6524  6533  6542  6743  6755  6841
                6843  6845  6847  6854  6859  6865  6961  6963  6965  6967
.MSOB           1840  1861  2098  2119  2682  2703  2906  2927  3084  3105  3276  3297  3468  3489
                3660  3681  3872  3893  4076  4097  4273  4294  4571  4592  4812  4833  5139  5160
                5422  5443  5655  5676
.MSAB           1844  1865  2102  2123  2329  2338  2349  2358  2367  2376  2385  2394  2403  2412
                2421  2430  2439  2448  2457  2466  2475  2484  2493  2502  2511  2520  2529  2538
                2547  2556  2565  2574  2583  2592  2601  2610  2619  2628  2637  2646  2655  2671
                2686  2707  2910  2931  3088  3109  3280  3301  3472  3493  3664  3685  3876  3897
                4080  4101  4246  4277  4298  4575  4596  4816  4837  5143  5164  5426  5447  5659
                5680  5818  6267  6276  6285  6294  6303  6312  6321  6330  6339  6348  6357  6366
                6375  6384  6393  6402  6411  6420  6429  6438  6447  6456  6465  6474  6483  6492
                6501  6510  6519  6528  6537  6546  6969
.MSDO           1838  1859  2096  2117  2327  2336  2347  2356  2365  2374  2383  2392  2401  2410
                2419  2428  2437  2446  2455  2464  2473  2482  2491  2500  2509  2518  2527  2536
                2545  2554  2563  2572  2581  2590  2599  2608  2617  2626  2635  2644  2653  2664
                2680  2701  2889  2904  2925  3082  3103  3229  3233  3274  3295  3421  3425  3466
```

```
                3487  3658  3679  3870  3891  4074  4095  4253  4262  4271  4292  4540  4560  4569
```

```
         3487   3658   3679   3870   3891   4074   4095   4253   4262   4271   4292   4540   4560   4569
         4590   4776   4810   4831   5058   5060   5062   5064   5066   5068   5070   5072   5079   5081
         5083   5085   5087   5089   5091   5093   5112   5114   5126   5128   5137   5158   5420   5441
         5606   5621   5653   5674   5813   5820   5946   6194   6196   6198   6200   6202   6204   6206
         6208   6210   6212   6214   6216   6218   6220   6222   6224   6226   6228   6230   6232   6234
         6236   6238   6240   6242   6244   6246   6248   6250   6252   6254   6256   6265   6269   6274
         6278   6283   6287   6292   6296   6301   6305   6310   6314   6319   6323   6328   6332   6337
         6341   6346   6350   6355   6359   6364   6368   6373   6377   6382   6386   6391   6395   6400
         6404   6409   6413   6418   6422   6427   6431   6436   6440   6445   6449   6454   6458   6463
         6467   6472   6476   6481   6485   6490   6494   6499   6503   6508   6512   6517   6521   6526
         6530   6535   6539   6544   6548   6741   6753   6852   6861   6863
.MSELC   1760   1771   1782   1788   1799   1805   1816   1831   4531   4542   4548   4558   4564   4778
         4790   4794   5326   5339   5349   5362   5368   5381   5389   5404   5413   5597   5608   5623
         5638
.MSELE   1229   1236   1433   1440   1447   1454   1461   1468   1475   1846   1848   1852   1867   1869
         1873   2035   2037   2044   2050   2059   2065   2067   2077   2085   2089   2104   2106   2110
         2125   2127   2131   2688   2690   2694   2709   2711   2715   2859   2863   2874   2878   2889
         2891   2895   2912   2914   2918   2933   2935   2939   3065   3069   3090   3092   3096   3111
         3113   3117   3282   3284   3288   3303   3305   3309   3474   3476   3480   3495   3497   3501
         3666   3668   3672   3687   3689   3693   3878   3880   3884   3899   3901   3905   4052   4054
         4063   4065   4082   4084   4088   4103   4105   4109   4279   4281   4285   4300   4302   4306
         4562   4577   4579   4583   4598   4600   4604   4818   4820   4824   4839   4841   4845   5145
         5147   5151   5166   5168   5172   5428   5430   5434   5449   5451   5455   5661   5663   5667
         5682   5684   5688   5815   5822   6181   6183   6185   6187   6755   6854   6865
.MSELF   5114   5128   5402
.MSELM   1330   1566   1573   1580   1758   1769   1780   1797   1814   1822   3231   3248   3423   3440
         3632   4509   4522   4533   4540   4560   4801   5100   5110   5124   5415   5644
.MSELP   3233   3425   3613   5112   5126
.MSKCN   4253   4262   6269   6278   6287   6296   6305   6314   6323   6332   6341   6350   6359   6368
         6377   6386   6395   6404   6413   6422   6431   6440   6449   6458   6467   6476   6485   6494
         6503   6512   6521   6530   6539   6548   6741
.MSKMB   6969
.MXNOR   2329   4246   5818   5950   5956   5962   5968   5974   5980   5986   5992   5998   6004   6010
         6016   6022   6028   6034   6040   6046   6052   6058   6064   6070   6076   6082   6088   6094
         6100   6106   6112   6118   6124   6130   6136   6142   6148   6154   6160   6166   6847
.MXOR    6267   6276   6285   6294   6303   6312   6321   6330   6339   6348   6357   6366   6375   6384
         6393   6402   6411   6420   6429   6438   6447   6456   6465   6474   6483   6492   6501   6510
         6519   6528   6537   6546
.RA      1844   1846   1852   1865   1867   1873   2102   2104   2110   2123   2125   2131   2686   2688
         2694   2707   2709   2715   2910   2912   2918   2931   2933   2939   3088   3090   3096   3109
         3111   3117   3231   3280   3282   3288   3301   3303   3309   3423   3472   3474   3480   3493
         3495   3501   3664   3666   3672   3685   3687   3693   3876   3878   3884   3897   3899   3905
         4080   4082   4088   4101   4103   4109   4277   4279   4285   4298   4300   4306   4575   4577
         4583   4596   4598   4604   4816   4818   4824   4837   4839   4845   5110   5124   5143   5145
         5151   5164   5166   5172   5426   5428   5434   5447   5449   5455   5659   5661   5667   5680
         5682   5688   5948   5950   5952   5954   5956   5958   5960   5962   5964   5966   5968   5970
         5972   5974   5976   5978   5980   5982   5984   5986   5988   5990   5992   5994   5996   5998
         6000   6002   6004   6006   6008   6010   6012   6014   6016   6018   6020   6022   6024   6026
         6028   6030   6032   6034   6036   6038   6040   6042   6044   6046   6048   6050   6052   6054
         6056   6058   6060   6062   6064   6066   6068   6070   6072   6074   6076   6078   6080   6082
         6084   6086   6088   6090   6092   6094   6096   6098   6100   6102   6104   6105   6108   6110
         6112   6114   6116   6118   6120   6122   6124   6126   6128   6130   6132   6134   6136   6138
         6140   6142   6144   6146   6148   6150   6152   6154   6156   6158   6160   6162   6164   6166
         6181   6183   6185   6187   6263   6267   6272   6276   6281   6285   6290   6294   6299   6303
         6308   6312   6317   6321   6326   6330   6335   6339   6344   6348   6353   6357   6362   6366
         6371   6375   6380   6384   6389   6393   6398   6402   6407   6411   6416   6420   6425   6429
         6434   6438   6443   6447   6452   6456   6461   6465   6470   6474   6479   6483   6488   6492
         6497   6501   6506   6510   6515   6519   6524   6528   6533   6537   6542   6546   6969
.RAND    1838   1859   2065   2096   2117   2327   2336   2347   2356   2365   2374   2383   2392   2401
         2410   2419   2428   2437   2446   2455   2464   2473   2482   2491   2500   2509   2518   2527
         2536   2545   2554   2563   2572   2581   2590   2599   2608   2617   2626   2635   2644   2653
         2664   2680   2701   2889   2904   2925   3082   3103   3274   3295   3466   3487   3658   3679
         3870   3891   4043   4052   4063   4074   4095   4235   4237   4244   4271   4292   4540   4569
         4590   4776   4810   4831   5137   5158   5402   5420   5441   5606   5621   5653   5674   5811
         5946   6841   6845   6859   6961
.RB      1838   1840   1844   1859   1861   1865   2096   2098   2102   2117   2119   2123   2680   2682
         2686   2701   2703   2707   2904   2906   2910   2925   2927   2931   3082   3084   3088   3103
         3105   3109   3229   3274   3276   3280   3295   3297   3301   3421   3466   3468   3472   3487
         3489   3493   3658   3660   3664   3679   3681   3685   3870   3872   3876   3891   3893   3897
         4074   4076   4080   4095   4097   4101   4271   4273   4277   4292   4294   4298   4569   4571
         4575   4590   4592   4596   4810   4812   4816   4831   4833   4837   5114   5128   5137   5139
         5143   5158   5160   5164   5420   5422   5426   5441   5443   5447   5653   5655   5659   5674
         5676   5680   5946   5948   5950   5952   5954   5956   5958   5960   5962   5964   5966   5968
         5970   5972   5974   5976   5978   5980   5982   5984   5986   5988   5990   5992   5994   5996
         5998   6000   6002   6004   6006   6008   6010   6012   6014   6016   6018   6020   6022   6024
         6026   6028   6030   6032   6034   6036   6038   6040   6042   6044   6046   6048   6050   6052
         6054   6056   6058   6060   6062   6064   6066   6068   6070   6072   6074   6076   6078   6080
         6082   6084   6086   6088   6090   6092   6094   6096   6098   6100   6102   6104   6106   6108
         6110   6112   6114   6116   6118   6120   6122   6124   6126   6128   6130   6132   6134   6136
         6138   6140   6142   6144   6146   6148   6150   6152   6154   6156   6158   6160   6162   6164
         6166   6194   6196   6198   6200   6202   6204   6206   6208   6210   6212   6214   6216   6218
         6220   6222   6224   6226   6228   6230   6232   6234   6236   6238   6240   6242   6244   6246
         6248   6250   6252   6254   6256   6265   6267   6269   6274   6276   6278   6283   6285   6287
         6292   6294   6296   6301   6303   6305   6310   6312   6314   6319   6321   6323   6328   6330
         6332   6337   6339   6341   6346   6348   6350   6355   6357   6359   6364   6366   6368   6373
         6375   6377   6382   6384   6386   6391   6393   6395   6400   6402   6404   6409   6411   6413
         6418   6420   6422   6427   6429   6431   6436   6438   6440   6445   6447   6449   6454   6456
         6458   6463   6465   6467   6472   6474   6476   6481   6483   6485   6490   6492   6494   6499
         6501   6503   6508   6510   6512   6517   6519   6521   6526   6528   6530   6535   6537   6539
         6544   6546   6548   6967
.RBAD    1217   1224   1231   1238   1332   1428   1435   1442   1449   1456   1463   1470   1477   1568
         1575   1582   6168   6173
.RCCAF   3237   3252   3259   3263   3267   3451   3455   3459   3636   3643   3647   3651
.RCCBF   3429   3444
.RCCCA   1751   1762   1773   1784   1790   1801   1807   1818   1824   4554   4780   5398   5610   5625
         5640
.RCCCC   2850   2865   2880   3071
.RCCCE   5330   5353   5372   5391
.RCCCP   4768   4784   4803
.RCCEB   3056   3075
.RCCER   2079   3841   3850   3859   3863
.RCCFZ   2329   2340   2664   2673   6267   6276   6285   6294   6303   6312   6321   6330   6339   6348
         6357   6366   6375   6384   6393   6402   6411   6420   6429   6438   6447   6456   6465   6474
         6483   6492   6501   6510   6519   6528   6537   6546
.RCCGC   1848   1869   2028   2037   2046   2052   2061   2069   2083   2106   2127   2690   2711   2859
         2874   2891   2897   2914   2935   3065   3092   3113   3284   3305   3476   3497   3668   3689
```

```
          3880  3901  4084  4105  4281  4302  4579  4600  4820  4841  5147  5168  5430  5451
          5663  5684
.RCCIA    4045  4056  4067
.RCCLW    5599  5614  5629  5646
.RCCMB    4237  4246  4255  4264
.RCCMP    4474  4486  4498  4511  4524
.RCCPP    5050  5102  5116  5130
.RCCXA    3621
.RCENA    1751  1762  1773  1784  1790  1801  1807  1818  1824  1848  1869  2028  2037  2046
          2052  2061  2069  2079  2083  2106  2127  2329  2340  2664  2673  2690  2711  2850
          2859  2865  2874  2880  2891  2897  2914  2935  3056  3065  3071  3075  3092  3113
          3237  3252  3259  3263  3267  3284  3305  3429  3444  3451  3455  3459  3476  3497
          3621  3636  3643  3647  3651  3668  3689  3841  3850  3859  3863  3880  3901  4045
          4056  4067  4084  4105  4237  4246  4255  4264  4281  4302  4474  4486  4498  4511
          4524  4554  4579  4600  4768  4780  4784  4803  4820  4841  5050  5102  5116  5130
          5147  5168  5330  5353  5372  5391  5398  5430  5451  5599  5610  5614  5625  5629
          5640  5646  5663  5684  6267  6276  6285  6294  6303  6312  6321  6330  6339  6348
          6357  6366  6375  6384  6393  6402  6411  6420  6429  6438  6447  6456  6465  6474
          6483  6492  6501  6510  6519  6528  6537  6546
.RCJP     1751  1762  1773  1784  1790  1801  1807  1818  1824  1848  1869  2028  2037  2046
          2052  2061  2069  2079  2083  2106  2127  2329  2340  2664  2673  2690  2711  2850
          2859  2865  2874  2880  2891  2897  2914  2935  3056  3065  3071  3075  3092  3113
          3237  3252  3259  3263  3267  3284  3305  3429  3444  3451  3455  3459  3476  3497
          3621  3636  3643  3647  3651  3668  3689  3841  3850  3859  3863  3880  3901  4045
          4056  4067  4084  4105  4237  4246  4255  4264  4281  4302  4474  4486  4498  4511
          4524  4554  4579  4600  4768  4780  4784  4803  4820  4841  5050  5102  5116  5130
          5147  5168  5330  5353  5372  5391  5398  5430  5451  5599  5610  5614  5625  5629
          5640  5646  5663  5684  6267  6276  6285  6294  6303  6312  6321  6330  6339  6348
          6357  6366  6375  6384  6393  6402  6411  6420  6429  6438  6447  6456  6465  6474
          6483  6492  6501  6510  6519  6528  6537  6546
.RCJS     3227  3246  3419  3438  3611  3630  4484  4496  4507  4520  5343  5345  5347  5366
          5385  5387  5948  5950  5954  5956  5960  5962  5966  5968  5972  5974  5978  5980
          5984  5986  5990  5992  5996  5998  6002  6004  6008  6010  6014  6016  6020  6022
          6026  6028  6032  6034  6038  6040  6044  6046  6050  6052  6056  6058  6062  6064
          6068  6070  6074  6076  6080  6082  6086  6088  6092  6094  6098  6100  6104  6106
          6110  6112  6116  6118  6122  6124  6128  6130  6134  6136  6140  6142  6146  6148
          6152  6154  6158  6160  6164  6166  6194  6196  6198  6200  6202  6204  6206  6208
          6210  6212  6214  6216  6218  6220  6222  6224  6226  6228  6230  6232  6234  6236
          6238  6240  6242  6244  6246  6248  6250  6252  6254  6256
.RCONT    1215  1222  1229  1236  1330  1426  1433  1440  1447  1454  1461  1468  1475  1566
          1573  1580  1760  1771  1782  1788  1799  1805  1816  1822  1838  1840  1846  1852
          1859  1861  1867  1873  2050  2065  2067  2085  2089  2096  2098  2104  2110  2117
          2119  2125  2131  2327  2336  2338  2347  2356  2365  2374  2383  2392  2401  2410
          2419  2428  2437  2446  2455  2464  2473  2482  2491  2500  2509  2518  2527  2536
          2545  2554  2563  2572  2581  2590  2599  2608  2617  2626  2635  2644  2653  2680
          2682  2688  2694  2701  2703  2709  2715  2863  2878  2889  2895  2904  2906  2912
          2918  2925  2927  2933  2939  3069  3082  3084  3090  3096  3103  3105  3111  3117
          3229  3231  3233  3235  3248  3250  3274  3276  3282  3288  3295  3297  3303  3309
          3421  3423  3425  3427  3440  3442  3466  3468  3474  3480  3487  3489  3495  3501
          3613  3615  3617  3619  3632  3634  3658  3660  3666  3672  3679  3681  3687  3693
          3870  3872  3878  3884  3891  3893  3899  3905  4054  4065  4074  4076  4082  4088
          4095  4097  4103  4109  4271  4273  4279  4285  4292  4294  4300  4306  4509  4522
          4542  4548  4558  4560  4562  4569  4571  4577  4583  4590  4592  4598  4604  4790
```

```
        4801  4810  4812  4818  4824  4831  4833  4839  4845  5100  5112  5114  5126  5128
        5137  5139  5145  5151  5158  5160  5166  5172  5389  5402  5420  5422  5428  5434
        5441  5443  5449  5455  5597  5608  5623  5638  5644  5653  5655  5661  5667  5674
        5676  5682  5688  5813  5815  5820  5822  5946  5952  5958  5964  5970  5976  5982
        5988  5994  6000  6006  6012  6018  6024  6030  6036  6042  6048  6054  6060  6066
        6072  6078  6084  6090  6096  6102  6108  6114  6120  6126  6132  6138  6144  6150
        6156  6162  6263  6265  6272  6274  6281  6283  6290  6292  6299  6301  6308  6310
        6317  6319  6326  6328  6335  6337  6344  6346  6353  6355  6362  6364  6371  6373
        6380  6382  6389  6391  6398  6400  6407  6409  6416  6418  6425  6427  6434  6436
        6443  6445  6452  6454  6461  6463  6470  6472  6479  6481  6488  6490  6497  6499
        6506  6508  6515  6517  6524  6526  6533  6535  6542  6544  6741  6743  6843  6845
        6847  6852  6854  6861  6863  6865  6967
.RCRTN  3261  3265  3269  3453  3457  3461  3645  3649  3653  4564  5408  6258  6551
.RCRY   1840  1861  2098  2119  2338  2349  2358  2367  2376  2385  2394  2403  2412  2421
        2430  2439  2448  2457  2466  2475  2484  2493  2502  2511  2520  2529  2538  2547
        2556  2565  2574  2583  2592  2601  2610  2619  2628  2637  2646  2655  2682  2703
        2906  2927  3084  3105  3276  3297  3468  3489  3660  3681  3872  3893  4076  4097
        4273  4294  4571  4592  4812  4833  5139  5160  5422  5443  5655  5676  5952  6963
.RD     1760  1762  1764  1771  1773  1775  1782  1784  1786  1788  1799  1801  1803  1805
        1816  1818  1820  1822  1838  1840  1844  1846  1848  1850  1852  1854  1859  1861
        1865  1867  1869  1871  1873  1875  2065  2096  2098  2102  2104  2106  2108  2110
        2112  2117  2119  2123  2125  2127  2129  2131  2133  2327  2329  2336  2338  2340
        2347  2349  2356  2358  2365  2367  2374  2376  2383  2385  2392  2394  2401  2403
        2410  2412  2419  2421  2428  2430  2437  2439  2446  2448  2455  2457  2464  2466
        2473  2475  2482  2484  2491  2493  2500  2502  2509  2511  2518  2520  2527  2529
        2536  2538  2545  2547  2554  2556  2563  2565  2572  2574  2581  2583  2590  2592
        2599  2601  2608  2610  2617  2619  2626  2628  2635  2637  2644  2646  2653  2655
        2657  2664  2671  2673  2680  2682  2686  2688  2690  2692  2694  2696  2701  2703
        2707  2709  2711  2713  2715  2717  2859  2861  2863  2874  2876  2878  2889  2891
        2893  2895  2904  2906  2910  2912  2914  2916  2918  2920  2925  2927  2931  2933
        2935  2937  2939  2941  3065  3067  3069  3082  3084  3088  3090  3092  3094  3096
        3098  3103  3105  3109  3111  3113  3115  3117  3119  3229  3231  3233  3235  3250
        3274  3276  3280  3282  3284  3286  3288  3290  3295  3297  3301  3303  3305  3307
        3309  3311  3421  3423  3425  3427  3442  3466  3468  3472  3474  3476  3478  3480
        3482  3487  3489  3493  3495  3497  3499  3501  3503  3613  3615  3617  3619  3634
        3658  3660  3664  3666  3668  3670  3672  3674  3679  3681  3685  3687  3689  3691
        3693  3695  3870  3872  3876  3878  3880  3882  3884  3886  3891  3893  3897  3899
        3901  3903  3905  3907  4043  4052  4063  4074  4076  4080  4082  4084  4086  4088
        4090  4095  4097  4101  4103  4105  4107  4109  4111  4235  4237  4244  4246  4253
        4255  4262  4264  4271  4273  4277  4279  4281  4283  4285  4287  4292  4294  4298
        4300  4302  4304  4306  4308  4540  4542  4544  4546  4548  4550  4552  4554  4556
        4558  4560  4562  4569  4571  4575  4577  4579  4581  4583  4585  4590  4592  4596
        4598  4600  4602  4604  4606  4776  4778  4780  4782  4810  4812  4816  4818  4820
        4822  4824  4826  4831  4833  4837  4839  4841  4843  4845  4847  5058  5060  5062
        5064  5066  5068  5070  5072  5079  5081  5083  5085  5087  5089  5091  5093  5110
        5112  5114  5124  5126  5128  5137  5139  5143  5145  5147  5149  5151  5153  5158
        5160  5164  5166  5168  5170  5172  5174  5343  5345  5347  5366  5385  5387  5398
        5400  5402  5404  5420  5422  5426  5428  5430  5432  5434  5436  5441  5443  5447
        5449  5451  5453  5455  5457  5606  5608  5610  5612  5621  5623  5625  5627  5638
        5640  5642  5653  5655  5659  5661  5663  5665  5667  5669  5674  5676  5680  5682
        5684  5686  5688  5690  5811  5813  5815  5818  5820  5822  5946  5948  5950  5952
        5954  5956  5958  5960  5962  5964  5966  5968  5970  5972  5974  5976  5978  5980
        5982  5984  5986  5988  5990  5992  5994  5996  5998  6000  6002  6004  6006  6008
```

```
87     6010    6012    6014    6016    6018    6020    6022    6024    6026    6028    6030    6032    6034    6036        SEQ 1600
       6038    6040    6042    6044    6046    6048    6050    6052    6054    6056    6058    6060    6062    6064
       6066    6068    6070    6072    6074    6076    6078    6080    6082    6084    6086    6088    6090    6092
       6094    6096    6098    6100    6102    6104    6106    6108    6110    6112    6114    6116    6118    6120
       6122    6124    6126    6128    6130    6132    6134    6136    6138    6140    6142    6144    6146    6148
       6150    6152    6154    6156    6158    6160    6162    6164    6166    6181    6183    6185    6187    6194
       6196    6198    6200    6202    6204    6206    6208    6210    6212    6214    6216    6218    6220    6222
       6224    6226    6228    6230    6232    6234    6236    6238    6240    6242    6244    6246    6248    6250
       6252    6254    6256    6263    6265    6267    6269    6272    6274    6276    6278    6281    6283    6285
       6287    6290    6292    6294    6296    6299    6301    6303    6305    6308    6310    6312    6314    6317
       6319    6321    6323    6326    6328    6330    6332    6335    6337    6339    6341    6344    6346    6348
       6350    6353    6355    6357    6359    6362    6364    6366    6368    6371    6373    6375    6377    6380
       6382    6384    6386    6389    6391    6393    6395    6398    6400    6402    6404    6407    6409    6411
       6413    6416    6418    6420    6422    6425    6427    6429    6431    6434    6436    6438    6440    6443
       6445    6447    6449    6452    6454    6456    6458    6461    6463    6465    6467    6470    6472    6474
       6476    6479    6481    6483    6485    6488    6490    6492    6494    6497    6499    6501    6503    6506
       6508    6510    6512    6515    6517    6519    6521    6524    6526    6528    6530    6533    6535    6537
       6539    6542    6544    6546    6548    6741    6743    6753    6755    6841    6843    6845    6847    6852
       6854    6859    6861    6863    6865    6961    6963    6965    6967    6969
.RJ    1213    1217    1220    1224    1227    1231    1234    1238    1328    1332    1424    1428    1431    1435
       1438    1442    1445    1449    1452    1456    1459    1463    1466    1470    1473    1477    1564    1568
       1571    1575    1578    1582    1749    1751    1753    1758    1762    1764    1769    1773    1775    1780
       1784    1786    1790    1792    1797    1801    1803    1807    1809    1814    1818    1820    1824    1826
       1831    1833    1842    1844    1848    1850    1854    1863    1865    1869    1871    1875    2026    2028
       2030    2035    2037    2039    2044    2046    2048    2052    2054    2059    2061    2063    2069    2071
       2077    2079    2081    2083    2087    2091    2100    2102    2106    2108    2112    2121    2123    2127
       2129    2133    2248    2250    2252    2254    2256    2258    2260    2262    2264    2266    2268    2270
       2272    2274    2276    2278    2280    2282    2284    2286    2288    2290    2292    2294    2296    2298
       2300    2302    2304    2306    2308    2310    2312    2314    2316    2318    2320    2322    2329    2331
       2340    2342    2349    2351    2358    2360    2367    2369    2376    2378    2385    2387    2394    2396
       2403    2405    2412    2414    2421    2423    2430    2432    2439    2441    2448    2450    2457    2459
       2466    2468    2475    2477    2484    2486    2493    2495    2502    2504    2511    2513    2520    2522
       2529    2531    2538    2540    2547    2549    2556    2558    2565    2567    2574    2576    2583    2585
       2592    2594    2601    2603    2610    2612    2619    2621    2628    2630    2637    2639    2646    2648
       2655    2657    2659    2664    2666    2671    2673    2675    2684    2686    2690    2692    2696    2705
       2707    2711    2713    2717    2848    2850    2852    2857    2859    2861    2865    2867    2872    2874
       2876    2880    2882    2887    2891    2893    2897    2899    2908    2910    2914    2916    2920    2929
       2931    2935    2937    2941    3054    3056    3058    3063    3065    3067    3071    3073    3075    3077
       3086    3088    3092    3094    3098    3107    3109    3113    3115    3119    3227    3237    3239    3244
       3246    3252    3254    3259    3263    3267    3278    3280    3284    3286    3290    3299    3301    3305
       3307    3311    3419    3429    3431    3436    3438    3444    3446    3451    3455    3459    3470    3472
       3476    3478    3482    3491    3493    3497    3499    3503    3611    3621    3623    3628    3630    3636
       3638    3643    3647    3651    3662    3664    3668    3670    3674    3683    3685    3689    3691    3695
       3839    3841    3843    3848    3850    3852    3857    3859    3861    3863    3865    3874    3876    3880
       3882    3886    3895    3897    3901    3903    3907    4043    4045    4047    4052    4056    4058    4063
       4067    4069    4078    4080    4084    4086    4090    4099    4101    4105    4107    4111    4235    4237
       4239    4244    4246    4248    4253    4255    4257    4262    4264    4266    4275    4277    4281    4283
       4287    4296    4298    4302    4304    4308    4472    4474    4476    4482    4484    4486    4488    4494
       4496    4498    4500    4505    4507    4511    4513    4518    4520    4524    4526    4531    4533    4535
       4540    4544    4546    4550    4552    4554    4556    4573    4575    4579    4581    4585    4594    4596
       4600    4602    4606    4766    4768    4770    4776    4778    4780    4782    4784    4786    4788    4792
       4794    4799    4803    4805    4814    4816    4820    4822    4826    4835    4837    4841    4843    4847
       5048    5050    5052    5058    5060    5062    5064    5066    5068    5070    5072    5079    5081    5083
```

```
88     5085    5087    5089    5091    5093    5098    5102    5104    5110    5116    5118    5124    5130    5132        SEQ 1601
       5141    5143    5147    5149    5153    5162    5164    5168    5170    5174    5324    5326    5328    5330
```

588

```
          5085  5087  5089  5091  5093  5098  5102  5104  5110  5116  5118  5124  5130  5132     SEQ 1601
          5141  5143  5147  5149  5153  5162  5164  5168  5170  5174  5324  5326  5328  5330
          5332  5337  5339  5341  5343  5345  5347  5349  5351  5353  5355  5360  5362  5364
          5366  5368  5370  5372  5374  5379  5381  5383  5385  5387  5391  5393  5398  5400
          5404  5406  5413  5415  5424  5426  5430  5432  5436  5445  5447  5451  5453  5457
          5595  5599  5601  5606  5610  5612  5614  5616  5621  5625  5627  5629  5631  5636
          5640  5642  5646  5648  5657  5659  5663  5665  5669  5678  5680  5684  5686  5690
          5811  5818  5948  5950  5954  5956  5960  5962  5966  5968  5972  5974  5978  5980
          5984  5986  5990  5992  5996  5998  6002  6004  6008  6010  6014  6016  6020  6022
          6026  6028  6032  6034  6038  6040  6044  6046  6050  6052  6056  6058  6062  6064
          6068  6070  6074  6076  6080  6082  6086  6088  6092  6094  6098  6100  6104  6106
          6110  6112  6116  6118  6122  6124  6128  6130  6134  6136  6140  6142  6146  6148
          6152  6154  6158  6160  6164  6166  6168  6173  6181  6183  6185  6187  6189  6194
          6196  6198  6200  6202  6204  6206  6208  6210  6212  6214  6216  6218  6220  6222
          6224  6226  6228  6230  6232  6234  6236  6238  6240  6242  6244  6246  6248  6250
          6252  6254  6256  6267  6269  6276  6278  6285  6287  6294  6296  6303  6305  6312
          6314  6321  6323  6330  6332  6339  6341  6348  6350  6357  6359  6366  6368  6375
          6377  6384  6386  6393  6395  6402  6404  6411  6413  6420  6422  6429  6431  6438
          6440  6447  6449  6456  6458  6465  6467  6474  6476  6483  6485  6492  6494  6501
          6503  6510  6512  6519  6521  6528  6530  6537  6539  6546  6548  6745  6753  6755
          6841  6859  6867  6961  6963  6965  6969
.RJMAP    1213  1217  1220  1224  1227  1231  1234  1238  1328  1332  1424  1428  1431  1435
          1438  1442  1445  1449  1452  1456  1459  1463  1466  1470  1473  1477  1564  1568
          1571  1575  1578  1582  1749  1753  1758  1764  1769  1775  1780  1786  1792  1797
          1803  1809  1814  1820  1826  1831  1833  1850  1854  1871  1875  2026  2030  2035
          2039  2044  2048  2054  2059  2063  2071  2077  2081  2087  2091  2108  2112  2129
          2133  2248  2250  2252  2254  2256  2258  2260  2262  2264  2266  2268  2270  2272
          2274  2276  2278  2280  2282  2284  2286  2288  2290  2292  2294  2296  2298  2300
          2302  2304  2306  2308  2310  2312  2314  2316  2318  2320  2322  2331  2342  2351
          2360  2369  2378  2387  2396  2405  2414  2423  2432  2441  2450  2459  2468  2477
          2486  2495  2504  2513  2522  2531  2540  2549  2558  2567  2576  2585  2594  2603
          2612  2621  2630  2639  2648  2659  2666  2675  2692  2696  2713  2717  2848  2852
          2857  2861  2867  2872  2876  2882  2887  2893  2899  2916  2920  2937  2941  3054
          3058  3063  3067  3073  3077  3094  3098  3115  3119  3239  3244  3254  3286  3290
          3307  3311  3431  3436  3446  3478  3482  3499  3503  3623  3628  3638  3670  3674
          3691  3695  3839  3843  3848  3852  3857  3861  3865  3882  3886  3903  3907  4043
          4047  4052  4058  4063  4069  4086  4090  4107  4111  4235  4239  4244  4248  4253
          4257  4262  4266  4283  4287  4304  4308  4472  4476  4482  4488  4494  4500  4505
          4513  4518  4526  4531  4533  4535  4540  4556  4581  4585  4602  4606  4766  4770
          4776  4782  4786  4792  4794  4799  4805  4822  4826  4843  4847  5048  5052  5058
          5060  5062  5064  5066  5068  5070  5072  5079  5081  5083  5085  5087  5089  5091
          5093  5098  5104  5110  5118  5124  5132  5149  5153  5170  5174  5324  5332  5337
          5355  5360  5374  5379  5393  5400  5413  5415  5432  5436  5453  5457  5595  5601
          5606  5612  5616  5621  5627  5631  5636  5642  5648  5665  5669  5686  5690  5811
          5818  6168  6173  6181  6183  6185  6187  6189  6269  6278  6287  6296  6305  6314
          6323  6332  6341  6350  6359  6368  6377  6386  6395  6404  6413  6422  6431  6440
          6449  6458  6467  6476  6485  6494  6503  6512  6521  6530  6539  6548  6745  6753
          6755  6841  6859  6867  6961  6969
.RJZ      3225  3417  3609  5941  6747
.RLDCT    1842  1863  2100  2121  2349  2358  2367  2376  2385  2394  2403  2412  2421  2430
          2439  2448  2457  2466  2475  2484  2493  2502  2511  2520  2529  2538  2547  2556
          2565  2574  2583  2592  2601  2610  2619  2628  2637  2646  2655  2684  2705  2908
          2929  3086  3107  3278  3299  3470  3491  3662  3683  3874  3895  4078  4099  4275
```

```
PNT
PNT
PNT
PNT
PNT

PNT
PUT
RGE
RPU
RTN
```

589

```
          4296  4544  4550  4573  4594  4778  4814  4835  5141  5162  5326  5339  5349  5362     SEQ 1602
          5368  5381  5404  5424  5445  5657  5678  6963
                                                          6317  6326  6335  6344  6353  6362
```

```
589             4296    4544    4550    4573    4594    4778    4814    4835    5141    5162    5326    5339    5349    5362        SEQ 1602
                5368    5381    5404    5424    5445    5657    5678    6963
        .RLDLM  5811    5818    6263    6272    6281    6290    6299    6308    6317    6326    6335    6344    6353    6362
                6371    6380    6389    6398    6407    6416    6425    6434    6443    6452    6461    6470    6479    6488
                6497    6506    6515    6524    6533    6542    6743    6843    6847
        .RLSAD  6194    6196    6198    6200    6202    6204    6206    6208    6210    6212    6214    6216    6218    6220
                6222    6224    6226    6228    6230    6232    6234    6236    6238    6240    6242    6244    6246    6248
                6250    6252    6254    6256    6841    6845    6859
        .RMGC   1229    1236    1330    1433    1440    1447    1454    1461    1468    1475    1566    1573    1580    1758
                1760    1769    1771    1780    1782    1788    1797    1799    1805    1814    1816    1822    1831    1846
                1848    1852    1867    1869    1873    2035    2037    2044    2050    2059    2065    2067    2077    2085
                2089    2104    2106    2110    2125    2127    2131    2688    2690    2694    2709    2711    2715    2859
                2863    2874    2878    2889    2891    2895    2912    2914    2918    2933    2935    2939    3065    3069
                3090    3092    3096    3111    3113    3117    3229    3231    3233    3248    3282    3284    3288    3303
                3305    3309    3421    3423    3425    3440    3474    3476    3480    3495    3497    3501    3613    3632
                3666    3668    3672    3687    3689    3693    3878    3880    3884    3899    3901    3905    4052    4054
                4063    4065    4082    4084    4088    4103    4105    4109    4253    4262    4279    4281    4285    4300
                4302    4306    4509    4522    4531    4533    4540    4542    4548    4558    4560    4562    4564    4577
                4579    4583    4598    4600    4604    4778    4790    4794    4801    4818    4820    4824    4839    4841
                4845    5058    5060    5062    5064    5066    5068    5070    5072    5079    5081    5083    5085    5087
                5089    5091    5093    5100    5110    5112    5114    5124    5126    5128    5145    5147    5151    5166
                5168    5172    5326    5339    5349    5362    5368    5381    5389    5402    5404    5413    5415    5428
                5430    5434    5449    5451    5455    5597    5608    5623    5638    5644    5661    5663    5667    5682
                5684    5688    5811    5813    5815    5818    5820    5822    6181    6183    6185    6187    6194    6196
                6198    6200    6202    6204    6206    6208    6210    6212    6214    6216    6218    6220    6222    6224
                6226    6228    6230    6232    6234    6236    6238    6240    6242    6244    6246    6248    6250    6252
                6254    6256    6263    6265    6269    6272    6274    6278    6281    6283    6287    6290    6292    6296
                6299    6301    6305    6308    6310    6314    6317    6319    6323    6326    6328    6332    6335    6337
                6341    6344    6346    6350    6353    6355    6359    6362    6364    6368    6371    6373    6377    6380
                6382    6386    6389    6391    6395    6398    6400    6404    6407    6409    6413    6416    6418    6422
                6425    6427    6431    6434    6436    6440    6443    6445    6449    6452    6454    6458    6461    6463
                6467    6470    6472    6476    6479    6481    6485    6488    6490    6494    6497    6499    6503    6506
                6508    6512    6515    6517    6521    6524    6526    6530    6533    6535    6539    6542    6544    6548
                6741    6743    6753    6755    6841    6845    6847    6852    6854    6859    6861    6863    6865
        .ROENA  1215    1426    1433    1440    1461    1468    1475    1566    1580    1846    1867    2104    2125    2329
                2340    2673    2688    2709    2889    2912    2933    3090    3111    3231    3282    3303    3423    3474
                3495    3666    3687    3878    3899    4052    4063    4082    4103    4279    4300    4562    4577    4598
                4818    4839    5110    5124    5145    5166    5402    5428    5449    5661    5682    5811    5815    5818
                5822    6181    6183    6185    6187    6263    6272    6281    6290    6299    6308    6317    6326    6335
                6344    6353    6362    6371    6380    6389    6398    6407    6416    6425    6434    6443    6452    6461
                6470    6479    6488    6497    6506    6515    6524    6533    6542    6743    6755    6843    6847    6854
                6865    6969
        .ROR    1846    1852    1867    1873    2104    2110    2125    2131    2340    2657    2673    2688    2694    2709
                2715    2863    2878    2895    2912    2918    2933    2939    3069    3090    3096    3111    3117    3229
                3231    3282    3288    3303    3309    3421    3423    3474    3480    3495    3501    3666    3672    3687
                3693    3878    3884    3899    3905    4082    4088    4103    4109    4253    4255    4262    4264    4279
                4285    4300    4306    4560    4562    4577    4583    4598    4604    4318    4824    4839    4845    5058
                5060    5062    5064    5066    5068    5070    5072    5079    5081    5083    5085    5087    5089    5091
                5093    5114    5128    5145    5151    5166    5172    5428    5434    5449    5455    5661    5667    5682
                5688    5813    5815    5820    5822    5948    5954    5958    5960    5964    5966    5970    5972    5976
                5978    5982    5984    5988    5990    5994    5996    6000    6002    6006    6008    6012    6014    6018
                6020    6024    6026    6030    6032    6036    6038    6042    6044    6048    6050    6054    6056    6060
                6062    6066    6068    6072    6074    6078    6080    6084    6086    6090    6092    6096    6098    6102
```

```
1590            6104    6108    6110    6114    6116    6120    6122    6126    6128    6132    6134    6138    6140    6144        SEQ 1603
```

```
        6104  6108  6110  6114  6116  6120  6122  6126  6128  6132  6134  6138  6140  6144
        6146  6150  6152  6156  6158  6162  6164  6194  6196  6198  6200  6202  6204  6206
        6208  6210  6212  6214  6216  6218  6220  6222  6224  6226  6228  6230  6232  6234
        6236  6238  6240  6242  6244  6246  6248  6250  6252  6254  6256  6263  6265  6269
        6272  6274  6278  6281  6283  6287  6290  6292  6296  6299  6301  6305  6308  6310
        6314  6317  6319  6323  6326  6328  6332  6335  6337  6341  6344  6346  6350  6353
        6355  6359  6362  6364  6368  6371  6373  6377  6380  6382  6386  6389  6391  6395
        6398  6400  6404  6407  6409  6413  6416  6418  6422  6425  6427  6431  6434  6436
        6440  6443  6445  6449  6452  6454  6458  6461  6463  6467  6470  6472  6476  6479
        6481  6485  6488  6490  6494  6497  6499  6503  6506  6508  6512  6515  6517  6521
        6524  6526  6530  6533  6535  6539  6542  6544  6548  6741  6743  6753  6755  6843
        6852  6854  6965  6967
.RPLUS  1840  1844  1861  1865  2098  2102  2119  2123  2338  2349  2358  2367  2376  2385
        2394  2403  2412  2421  2430  2439  2448  2457  2466  2475  2484  2493  2502  2511
        2520  2529  2538  2547  2556  2565  2574  2583  2592  2601  2610  2619  2628  2637
        2646  2655  2671  2682  2686  2703  2707  2906  2910  2927  2931  3084  3088  3105
        3109  3276  3280  3297  3301  3468  3472  3489  3493  3660  3664  3681  3685  3872
        3876  3893  3897  4076  4080  4097  4101  4273  4277  4294  4298  4571  4575  4592
        4596  4812  4816  4833  4837  5139  5143  5160  5164  5422  5426  5443  5447  5655
        5659  5676  5680  5952  6963  6969
.RRAM   6263  6265  6272  6274  6281  6283  6290  6292  6299  6301  6308  6310  6317  6319
        6326  6328  6335  6337  6344  6346  6353  6355  6362  6364  6371  6373  6380  6382
        6389  6391  6398  6400  6407  6409  6416  6418  6425  6427  6434  6436  6443  6445
        6452  6454  6461  6463  6470  6472  6479  6481  6488  6490  6497  6499  6506  6508
        6515  6517  6524  6526  6533  6535  6542  6544  6852  6863
.RRDLM  1222  1426  1447  1454  1461  1468  1475  1573  1580  5813  5820  6265  6274  6283
        6292  6301  6310  6319  6328  6337  6346  6355  6364  6373  6382  6391  6400  6409
        6418  6427  6436  6445  6454  6463  6472  6481  6490  6499  6508  6517  6526  6535
        6544  6753  6852  6861  6863
.RRPCT  1844  1865  2102  2123  2657  2671  2686  2707  2910  2931  3088  3109  3280  3301
        3472  3493  3664  3685  3876  3897  4080  4101  4277  4298  4546  4552  4575  4596
        4788  4816  4837  5143  5164  5328  5341  5351  5364  5370  5383  5406  5426  5447
        5659  5680  6965
.RSOA   1846  1852  1867  1873  2065  2104  2110  2125  2131  2340  2657  2673  2688  2694
        2709  2715  2863  2878  2895  2912  2918  2933  2939  3069  3090  3096  3111  3117
        3231  3282  3288  3303  3309  3423  3474  3480  3495  3501  3666  3672  3687  3693
        3878  3884  3899  3905  4043  4052  4063  4082  4088  4103  4109  4235  4237  4244
        4255  4264  4279  4285  4300  4306  4562  4577  4583  4598  4604  4818  4824  4839
        4845  5110  5124  5145  5151  5166  5172  5402  5428  5434  5449  5455  5661  5667
        5682  5688  5811  5815  5822  5948  5950  5952  5954  5956  5958  5960  5962  5964
        5966  5968  5970  5972  5974  5976  5978  5980  5982  5984  5986  5988  5990  5992
        5994  5996  5998  6000  6002  6004  6006  6008  6010  6012  6014  6016  6018  6020
        6022  6024  6026  6028  6030  6032  6034  6036  6038  6040  6042  6044  6046  6048
        6050  6052  6054  6056  6058  6060  6062  6064  6066  6068  6070  6072  6074  6076
        6078  6080  6082  6084  6086  6088  6090  6092  6094  6096  6098  6100  6102  6104
        6106  6108  6110  6112  6114  6116  6118  6120  6122  6124  6126  6128  6130  6132
        6134  6136  6138  6140  6142  6144  6146  6148  6150  6152  6154  6156  6158  6160
        6162  6164  6166  6181  6183  6185  6187  6263  6272  6281  6290  6299  6308  6317
        6326  6335  6344  6353  6362  6371  6380  6389  6398  6407  6416  6425  6434  6443
        6452  6461  6470  6479  6488  6497  6506  6515  6524  6533  6542  6743  6755  6841
        6843  6845  6847  6854  6859  6865  6961  6963  6965  6967
.RSOB   1840  1861  2098  2119  2682  2703  2906  2927  3084  3105  3276  3297  3468  3489
        3660  3681  3872  3893  4076  4097  4273  4294  4571  4592  4812  4833  5139  5160
```

```
        5422  5443  5655  5676
```

```
              5422 5443 5655 5676
591 .RSAB     1844 1865 2102 2123 2329 2338 2349 2358 2367 2376 2385 2394 2403 2412     SEQ 1604    DFP
              2421 2430 2439 2448 2457 2466 2475 2484 2493 2502 2511 2520 2529 2538                 DFP
              2547 2556 2565 2574 2583 2592 2601 2610 2619 2628 2637 2646 2655 2671
              2686 2707 2910 2931 3088 3109 3280 3301 3472 3493 3664 3685 3876 3897
              4080 4101 4246 4277 4298 4575 4596 4816 4837 5143 5164 5426 5447 5659
              5680 5818 6267 6276 6285 6294 6303 6312 6321 6330 6339 6348 6357 6366
              6375 6384 6393 6402 6411 6420 6429 6438 6447 6456 6465 6474 6483 6492
              6501 6510 6519 6528 6537 6546 6969
    .RSDO     1838 1859 2096 2117 2327 2336 2347 2356 2365 2374 2383 2392 2401 2410
              2419 2428 2437 2446 2455 2464 2473 2482 2491 2500 2509 2518 2527 2536
              2545 2554 2563 2572 2581 2590 2599 2608 2617 2626 2635 2644 2653 2664
              2680 2701 2889 2904 2925 3082 3103 3229 3233 3274 3295 3421 3425 3466
              3487 3658 3679 3870 3891 4074 4095 4253 4262 4271 4292 4540 4560 4569
              4590 4776 4810 4831 5058 5060 5062 5064 5066 5068 5070 5072 5079 5081
              5083 5085 5087 5089 5091 5093 5112 5114 5126 5128 5137 5158 5420 5441
              5606 5621 5653 5674 5813 5820 5946 6194 6196 6198 6200 6202 6204 6206
              6208 6210 6212 6214 6216 6218 6220 6222 6224 6226 6228 6230 6232 6234
              6236 6238 6240 6242 6244 6246 6248 6250 6252 6254 6256 6265 6269 6274
              6278 6283 6287 6292 6296 6301 6305 6310 6314 6319 6323 6328 6332 6337
              6341 6346 6350 6355 6359 6364 6368 6373 6377 6382 6386 6391 6395 6400
              6404 6409 6413 6418 6422 6427 6431 6436 6440 6445 6449 6454 6458 6463
              6467 6472 6476 6481 6485 6490 6494 6499 6503 6508 6512 6517 6521 6526
              6530 6535 6539 6544 6548 6741 6753 6852 6861 6863
    .RSELC    1760 1771 1782 1788 1799 1805 1816 1831 4531 4542 4548 4558 4564 4778
              4790 4794 5326 5339 5349 5362 5368 5381 5389 5404 5413 5597 5608 5623
              5638
    .RSELE    1229 1236 1433 1440 1447 1454 1461 1468 1475 1846 1848 1852 1867 1869
              1873 2035 2037 2044 2050 2059 2065 2067 2077 2085 2089 2104 2106 2110
              2125 2127 2131 2688 2690 2694 2709 2711 2715 2859 2863 2874 2878 2889
              2891 2895 2912 2914 2918 2933 2935 2939 3065 3069 3090 3092 3096 3111
              3113 3117 3282 3284 3288 3303 3305 3309 3474 3476 3480 3495 3497 3501
              3666 3668 3672 3687 3689 3693 3878 3880 3884 3899 3901 3905 4052 4054
              4063 4065 4082 4084 4088 4103 4105 4109 4279 4281 4285 4300 4302 4306
              4562 4577 4579 4583 4598 4600 4604 4818 4820 4824 4839 4841 4845 5145
              5147 5151 5166 5168 5172 5428 5430 5434 5449 5451 5455 5661 5663 5667
              5682 5684 5688 5815 5822 6181 6183 6185 6187 6755 6854 6865
    .RSELF    5114 5128 5402
    .RSELM    1330 1566 1573 1580 1758 1769 1780 1797 1814 1822 3231 3248 3423 3440
              3632 4509 4522 4533 4540 4560 4801 5100 5110 5124 5415 5644
    .RSELP    3233 3425 3613 5112 5126
    .RSKCN    4253 4262 6269 6278 6287 6296 6305 6314 6323 6332 6341 6350 6359 6368
              6377 6386 6395 6404 6413 6422 6431 6440 6449 6458 6467 6476 6485 6494
              6503 6512 6521 6530 6539 6548 6741
    .RSKMB    6969
    .RXNOR    2329 4246 5818 5950 5956 5962 5968 5974 5980 5986 5992 5998 6004 6010
              6016 6022 6028 6034 6040 6046 6052 6058 6064 6070 6076 6082 6088 6094
              6100 6106 6112 6118 6124 6130 6136 6142 6148 6154 6160 6166 6847
    .RXOR     6267 6276 6285 6294 6303 6312 6321 6330 6339 6348 6357 6366 6375 6384
              6393 6402 6411 6420 6429 6438 6447 6456 6465 6474 6483 6492 6501 6510
              6519 6528 6537 6546
```

```
1592 ATABLE   5763 5764 5765 5767 5769 5770 6622 6623 6624 6626 6627 6825 6827 6829     SEQ 1605    DFP
```

```
1592

ATABLE   5763 5764 5765 5767 5769 5770 6622 6623 6624 6626 6627 6825 6827 6829   SEQ 1605
CALC     1213 1215 1217 1220 1222 1224 1227 1229 1231 1234 1236 1238 1328 1330
         1332 1424 1426 1428 1431 1433 1435 1438 1440 1442 1445 1447 1449 1452
         1454 1456 1459 1461 1463 1466 1468 1470 1473 1475 1477 1564 1566 1568
         1571 1573 1575 1578 1580 1582 1749 1751 1753 1758 1760 1762 1764 1769
         1771 1773 1775 1780 1782 1784 1786 1788 1790 1792 1797 1799 1801 1803
         1805 1807 1809 1814 1816 1818 1820 1822 1824 1826 1831 1833 1838 1840
         1842 1844 1846 1848 1850 1852 1854 1859 1861 1863 1865 1867 1869 1871
         1873 1875 2026 2028 2030 2035 2037 2039 2044 2046 2048 2050 2052 2054
         2059 2061 2063 2065 2067 2069 2071 2077 2079 2081 2083 2085 2087 2089
         2091 2096 2098 2100 2102 2104 2106 2108 2110 2112 2117 2119 2121 2123
         2125 2127 2129 2131 2133 2248 2250 2252 2254 2256 2258 2260 2262 2264
         2266 2268 2270 2272 2274 2276 2278 2280 2282 2284 2286 2288 2290 2292
         2294 2296 2298 2300 2302 2304 2306 2308 2310 2312 2314 2316 2318 2320
         2322 2327 2329 2331 2336 2338 2340 2342 2347 2349 2351 2356 2358 2360
         2365 2367 2369 2374 2376 2378 2383 2385 2387 2392 2394 2396 2401 2403
         2405 2410 2412 2414 2419 2421 2423 2428 2430 2432 2437 2439 2441 2446
         2448 2450 2455 2457 2459 2464 2466 2468 2473 2475 2477 2482 2484 2486
         2491 2493 2495 2500 2502 2504 2509 2511 2513 2518 2520 2522 2527 2529
         2531 2536 2538 2540 2545 2547 2549 2554 2556 2558 2563 2565 2567 2572
         2574 2576 2581 2583 2585 2590 2592 2594 2599 2601 2603 2608 2610 2612
         2617 2619 2621 2626 2628 2630 2635 2637 2639 2644 2646 2648 2653 2655
         2657 2659 2664 2666 2671 2673 2675 2680 2682 2684 2686 2688 2690 2692
         2694 2696 2701 2703 2705 2707 2709 2711 2713 2715 2717 2848 2850 2852
         2857 2859 2861 2863 2865 2867 2872 2874 2876 2878 2880 2882 2887 2889
         2891 2893 2895 2897 2899 2904 2906 2908 2910 2912 2914 2916 2918 2920
         2925 2927 2929 2931 2933 2935 2937 2939 2941 3054 3056 3058 3063 3065
         3067 3069 3071 3073 3075 3077 3082 3084 3086 3088 3090 3092 3094 3096
         3098 3103 3105 3107 3109 3111 3113 3115 3117 3119 3225 3227 3229 3231
         3233 3235 3237 3239 3244 3246 3248 3250 3252 3254 3259 3261 3263 3265
         3267 3269 3274 3276 3278 3280 3282 3284 3286 3288 3290 3295 3297 3299
         3301 3303 3305 3307 3309 3311 3417 3419 3421 3423 3425 3427 3429 3431
         3436 3438 3440 3442 3444 3446 3451 3453 3455 3457 3459 3461 3466 3468
         3470 3472 3474 3476 3478 3480 3482 3487 3489 3491 3493 3495 3497 3499
         3501 3503 3609 3611 3613 3615 3617 3619 3621 3623 3628 3630 3632 3634
         3636 3638 3643 3645 3647 3649 3651 3653 3658 3660 3662 3664 3666 3668
         3670 3672 3674 3679 3681 3683 3685 3687 3689 3691 3693 3695 3839 3841
         3843 3848 3850 3852 3857 3859 3861 3863 3865 3870 3872 3874 3876 3878
         3880 3882 3884 3886 3891 3893 3895 3897 3899 3901 3903 3905 3907 4043
         4045 4047 4052 4054 4056 4058 4063 4065 4067 4069 4074 4076 4078 4080
         4082 4084 4086 4088 4090 4095 4097 4099 4101 4103 4105 4107 4109 4111
         4235 4237 4239 4244 4246 4248 4253 4255 4257 4262 4264 4266 4271 4273
         4275 4277 4279 4281 4283 4285 4287 4292 4294 4296 4298 4300 4302 4304
         4306 4308 4472 4474 4476 4482 4484 4486 4488 4494 4496 4498 4500 4505
         4507 4509 4511 4513 4518 4520 4522 4524 4526 4531 4533 4535 4540 4542
         4544 4546 4548 4550 4552 4554 4556 4558 4560 4562 4564 4569 4571 4573
         4575 4577 4579 4581 4583 4585 4590 4592 4594 4596 4598 4600 4602 4604
         4606 4766 4768 4770 4776 4778 4780 4782 4784 4786 4788 4790 4792 4794
         4799 4801 4803 4805 4810 4812 4814 4816 4818 4820 4822 4824 4826 4831
         4833 4835 4837 4839 4841 4843 4845 4847 5048 5050 5052 5058 5060 5062
         5064 5066 5068 5070 5072 5079 5081 5083 5085 5087 5089 5091 5093 5098
         5100 5102 5104 5110 5112 5114 5116 5118 5124 5126 5128 5130 5132 5137
         5139 5141 5143 5145 5147 5149 5151 5153 5158 5160 5162 5164 5166 5168

1593

         5170 5172 5174 5324 5326 5328 5330 5332 5337 5339 5341 5343 5345 5347   SEQ 1606
```

```
1593

          5170  5172  5174  5324  5326  5328  5330  5332  5337  5339  5341  5343  5345  5347      SEQ 1606
          5349  5351  5353  5355  5360  5362  5364  5366  5368  5370  5372  5374  5379  5381
          5383  5385  5387  5389  5391  5393  5398  5400  5402  5404  5406  5408  5413  5415
          5420  5422  5424  5426  5428  5430  5432  5434  5436  5441  5443  5445  5447  5449
          5451  5453  5455  5457  5595  5597  5599  5601  5606  5608  5610  5612  5614  5616
          5621  5623  5625  5627  5629  5631  5636  5638  5640  5642  5644  5646  5648  5653
          5655  5657  5659  5661  5663  5665  5667  5669  5674  5676  5678  5680  5682  5684
          5686  5688  5690  5811  5813  5815  5818  5820  5822  5941  5946  5948  5950  5952
          5954  5956  5958  5960  5962  5964  5966  5968  5970  5972  5974  5976  5978  5980
          5982  5984  5986  5988  5990  5992  5994  5996  5998  6000  6002  6004  6006  6008
          6010  6012  6014  6016  6018  6020  6022  6024  6026  6028  6030  6032  6034  6036
          6038  6040  6042  6044  6046  6048  6050  6052  6054  6056  6058  6060  6062  6064
          6066  6068  6070  6072  6074  6076  6078  6080  6082  6084  6086  6088  6090  6092
          6094  6096  6098  6100  6102  6104  6106  6108  6110  6112  6114  6116  6118  6120
          6122  6124  6126  6128  6130  6132  6134  6136  6138  6140  6142  6144  6146  6148
          6150  6152  6154  6156  6158  6160  6162  6164  6166  6168  6173  6181  6183  6185
          6187  6189  6194  6196  6198  6200  6202  6204  6206  6208  6210  6212  6214  6216
          6218  6220  6222  6224  6226  6228  6230  6232  6234  6236  6238  6240  6242  6244
          6246  6248  6250  6252  6254  6256  6258  6263  6265  6267  6269  6272  6274  6276
          6278  6281  6283  6285  6287  6290  6292  6294  6296  6299  6301  6303  6305  6308
          6310  6312  6314  6317  6319  6321  6323  6326  6328  6330  6332  6335  6337  6339
          6341  6344  6346  6348  6350  6353  6355  6357  6359  6362  6364  6366  6368  6371
          6373  6375  6377  6380  6382  6384  6386  6389  6391  6393  6395  6398  6400  6402
          6404  6407  6409  6411  6413  6416  6418  6420  6422  6425  6427  6429  6431  6434
          6436  6438  6440  6443  6445  6447  6449  6452  6454  6456  6458  6461  6463  6465
          6467  6470  6472  6474  6476  6479  6481  6483  6485  6488  6490  6492  6494  6497
          6499  6501  6503  6506  6508  6510  6512  6515  6517  6519  6521  6524  6526  6528
          6530  6533  6535  6537  6539  6542  6544  6546  6548  6551  6741  6743  6745  6747
          6753  6755  6841  6843  6845  6847  6852  6854  6859  6861  6863  6865  6867  6961
          6963  6965  6967  6969
CONCAT    1213  1215  1217  1220  1222  1224  1227  1229  1231  1234  1236  1238  1328  1330
          1332  1424  1426  1428  1431  1433  1435  1438  1440  1442  1445  1447  1449  1452
          1454  1456  1459  1461  1463  1466  1468  1470  1473  1475  1477  1564  1566  1568
          1571  1573  1575  1578  1580  1582  1749  1751  1753  1758  1760  1762  1764  1769
          1771  1773  1775  1780  1782  1784  1786  1788  1790  1792  1797  1799  1801  1803
          1805  1807  1809  1814  1816  1818  1820  1822  1824  1826  1831  1833  1838  1840
          1842  1844  1846  1848  1850  1852  1854  1859  1861  1863  1865  1867  1869  1871
          1873  1875  2026  2028  2030  2035  2037  2039  2044  2046  2048  2050  2052  2054
          2059  2061  2063  2065  2067  2069  2071  2077  2079  2081  2083  2085  2087  2089
          2091  2096  2098  2100  2102  2104  2106  2108  2110  2112  2117  2119  2121  2123
          2125  2127  2129  2131  2133  2248  2250  2252  2254  2256  2258  2260  2262  2264
          2266  2268  2270  2272  2274  2276  2278  2280  2282  2284  2286  2288  2290  2292
          2294  2296  2298  2300  2302  2304  2306  2308  2310  2312  2314  2316  2318  2320
          2322  2327  2329  2331  2336  2338  2340  2342  2347  2349  2351  2356  2358  2360
          2365  2367  2369  2374  2376  2378  2383  2385  2387  2392  2394  2396  2401  2403
          2405  2410  2412  2414  2419  2421  2423  2428  2430  2432  2437  2439  2441  2446
          2448  2450  2455  2457  2459  2464  2466  2468  2473  2475  2477  2482  2484  2486
          2491  2493  2495  2500  2502  2504  2509  2511  2513  2518  2520  2522  2527  2529
          2531  2536  2538  2540  2545  2547  2549  2554  2556  2558  2563  2565  2567  2572
          2574  2576  2581  2583  2585  2590  2592  2594  2599  2601  2603  2608  2610  2612
          2617  2619  2621  2626  2628  2630  2635  2637  2639  2644  2646  2648  2653  2655
          2657  2659  2664  2666  2671  2673  2675  2680  2682  2684  2686  2688  2690  2692
          2694  2696  2701  2703  2705  2707  2709  2711  2713  2715  2717  2848  2850  2852

1594      2857  2859  2861  2863  2865  2867  2872  2874  2876  2878  2880  2882  2887  2889      SEQ 1607
```

1594

SEQ 1607

2857	2859	2861	2863	2865	2867	2872	2874	2876	2878	2880	2882	2887	2889
2891	2893	2895	2897	2899	2904	2906	2908	2910	2912	2914	2916	2918	2920
2925	2927	2929	2931	2933	2935	2937	2939	2941	3054	3056	3058	3063	3065
3067	3069	3071	3073	3075	3077	3082	3084	3086	3088	3090	3092	3094	3096
3098	3103	3105	3107	3109	3111	3113	3115	3117	3119	3225	3227	3229	3231
3233	3235	3237	3239	3244	3246	3248	3250	3252	3254	3259	3261	3263	3265
3267	3269	3274	3276	3278	3280	3282	3284	3286	3288	3290	3295	3297	3299
3301	3303	3305	3307	3309	3311	3417	3419	3421	3423	3425	3427	3429	3431
3436	3438	3440	3442	3444	3446	3451	3453	3455	3457	3459	3461	3466	3468
3470	3472	3474	3476	3478	3480	3482	3487	3489	3491	3493	3495	3497	3499
3501	3503	3609	3611	3613	3615	3617	3619	3621	3623	3628	3630	3632	3634
3636	3638	3643	3645	3647	3649	3651	3653	3658	3660	3662	3664	3666	3668
3670	3672	3674	3679	3681	3683	3685	3687	3689	3691	3693	3695	3839	3841
3843	3848	3850	3852	3857	3859	3861	3863	3865	3870	3872	3874	3876	3878
3880	3882	3894	3886	3891	3893	3895	3897	3899	3901	3903	3905	3907	4043
4045	4047	4052	4054	4056	4058	4063	4065	4067	4069	4074	4076	4078	4080
4082	4084	4086	4088	4090	4095	4097	4099	4101	4103	4105	4107	4109	4111
4235	4237	4239	5244	4246	4248	4253	4255	4257	4262	4264	4266	4271	4273
4275	4277	4279	4281	4283	4285	4287	4292	4294	4296	4298	4300	4302	4304
4306	4308	4472	4474	4476	4482	4484	4486	4488	4494	4496	4498	4500	4505
4507	4509	4511	4513	4518	4520	4522	4524	4526	4531	4533	4535	4540	4542
4544	4546	4548	4550	4552	4554	4556	4558	4560	4562	4564	4569	4571	4573
4575	4577	4579	4581	4583	4585	4590	4592	4594	4596	4598	4600	4602	4604
4606	4766	4768	4770	4776	4778	4780	4782	4784	4786	4788	4790	4792	4794
4799	4801	4803	4805	4810	4812	4814	4816	4818	4820	4822	4824	4826	4831
4833	4835	4837	4839	4841	4843	4845	4847	5048	5050	5052	5058	5060	5062
5064	5066	5068	5070	5072	5079	5081	5083	5085	5087	5089	5091	5093	5098
5100	5102	5104	5110	5112	5114	5116	5118	5124	5126	5128	5130	5132	5137
5139	5141	5143	5145	5147	5149	5151	5153	5158	5160	5162	5164	5166	5168
5170	5172	5174	5324	5326	5328	5330	5332	5337	5339	5341	5343	5345	5347
5349	5351	5353	5355	5360	5362	5364	5366	5368	5370	5372	5374	5379	5381
5383	5385	5387	5389	5391	5393	5398	5400	5402	5404	5406	5408	5413	5415
5420	5422	5424	5426	5428	5430	5432	5434	5436	5441	5443	5445	5447	5449
5451	5453	5455	5457	5595	5597	5599	5601	5606	5608	5610	5612	5614	5616
5621	5623	5625	5627	5629	5631	5636	5638	5640	5642	5644	5646	5648	5653
5655	5657	5659	5661	5663	5665	5667	5669	5674	5676	5678	5680	5682	5684
5686	5688	5690	5811	5813	5815	5818	5820	5822	5941	5946	5948	5950	5952
5954	5956	5958	5960	5962	5964	5966	5968	5970	5972	5974	5976	5978	5980
5982	5984	5986	5988	5990	5992	5994	5996	5998	6000	6002	6004	6006	6008
6010	6012	6014	6016	6018	6020	6022	6024	6026	6028	6030	6032	6034	6036
6038	6040	6042	6044	6046	6048	6050	6052	6054	6056	6058	6060	6062	6064
6066	6068	6070	6072	6074	6076	6078	6080	6082	6084	6086	6088	6090	6092
6094	6096	6098	6100	6102	6104	6106	6108	6110	6112	6114	6116	6118	6120
6122	6124	6126	6128	6130	6132	6134	6136	6138	6140	6142	6144	6146	6148
6150	6152	6154	6156	6158	6160	6162	6164	6166	6168	6173	6181	6183	6185
6187	6189	6194	6196	6198	6200	6202	6204	6206	6208	6210	6212	6214	6216
6218	6220	6222	6224	6226	6228	6230	6232	6234	6236	6238	6240	6242	6244
6246	6248	6250	6252	6254	6256	6258	6263	6265	6267	6269	6272	6274	6276
6278	6281	6283	6285	6287	6290	6292	6294	6296	6299	6301	6303	6305	6308
6310	6312	6314	6317	6319	6321	6323	6326	6328	6330	6332	6335	6337	6339
6341	6344	6346	6348	6350	6353	6355	6357	6359	6362	6364	6365	6368	6371
6373	6375	6377	6380	6382	6384	6386	6389	6391	6393	6395	6398	6400	6402
6404	6407	6409	6411	6413	6416	6418	6420	6422	6425	6427	6429	6431	6434

1595

SEQ 1608

6436	6438	6440	6443	6445	6447	6449	6452	6454	6456	6458	6461	6463	6465

```
1595          6436 6438 6440 6443 6445 6447 6449 6452 6454 6456 6458 6461 6463 6465   SEQ 1608
              6467 6470 6472 6474 6476 6479 6481 6483 6485 6488 6490 6492 6494 6497
              6499 6501 6503 6506 6508 6510 6512 6515 6517 6519 6521 6524 6526 6528
              6530 6533 6535 6537 6539 6542 6544 6546 6548 6551 6741 6743 6745 6747
              6753 6755 6841 6843 6845 6847 6852 6854 6859 6861 6863 6865 6867 6961
              6963 6965 6967 6969
       FIELD  1213 1215 1217 1220 1222 1224 1227 1229 1231 1234 1236 1238 1328 1330
              1332 1424 1426 1428 1431 1433 1435 1438 1440 1442 1445 1447 1449 1452
              1454 1456 1459 1461 1463 1466 1468 1470 1473 1475 1477 1564 1566 1568
              1571 1573 1575 1578 1580 1582 1749 1751 1753 1758 1760 1762 1764 1769
              1771 1773 1775 1780 1782 1784 1786 1788 1790 1792 1797 1799 1801 1803
              1805 1807 1809 1814 1816 1818 1820 1822 1824 1826 1831 1833 1838 1840
              1842 1844 1846 1848 1850 1852 1854 1859 1861 1863 1865 1867 1869 1871
              1873 1875 2026 2028 2030 2035 2037 2039 2044 2046 2048 2050 2052 2054
              2059 2061 2063 2065 2067 2069 2071 2077 2079 2081 2083 2085 2087 2089
              2091 2096 2098 2100 2102 2104 2106 2108 2110 2112 2117 2119 2121 2123
              2125 2127 2129 2131 2133 2248 2250 2252 2254 2256 2258 2260 2262 2264
              2266 2268 2270 2272 2274 2276 2278 2280 2282 2284 2286 2288 2290 2292
              2294 2296 2298 2300 2302 2304 2306 2308 2310 2312 2314 2316 2318 2320
              2322 2327 2329 2331 2336 2338 2340 2342 2347 2349 2351 2356 2358 2360
              2365 2367 2369 2374 2376 2378 2383 2385 2387 2392 2394 2396 2401 2403
              2405 2410 2412 2414 2419 2421 2423 2428 2430 2432 2437 2439 2441 2446
              2448 2450 2455 2457 2459 2464 2466 2468 2473 2475 2477 2482 2484 2486
              2491 2493 2495 2500 2502 2504 2509 2511 2513 2518 2520 2522 2527 2529
              2531 2536 2538 2540 2545 2547 2549 2554 2556 2558 2563 2565 2567 2572
              2574 2576 2581 2583 2585 2590 2592 2594 2599 2601 2603 2608 2610 2612
              2617 2619 2621 2626 2628 2630 2635 2637 2639 2644 2646 2648 2653 2655
              2657 2659 2664 2666 2671 2673 2675 2680 2682 2684 2686 2688 2690 2692
              2694 2696 2701 2703 2705 2707 2709 2711 2713 2715 2717 2848 2850 2852
              2857 2859 2861 2863 2865 2867 2872 2874 2876 2878 2880 2882 2887 2889
              2891 2893 2895 2897 2899 2904 2906 2908 2910 2912 2914 2916 2918 2920
              2925 2927 2929 2931 2933 2935 2937 2939 2941 3054 3056 3058 3063 3065
              3067 3069 3071 3073 3075 3077 3082 3084 3086 3088 3090 3092 3094 3096
              3098 3103 3105 3107 3109 3111 3113 3115 3117 3119 3225 3227 3229 3231
              3233 3235 3237 3239 3244 3246 3248 3250 3252 3254 3259 3261 3263 3265
              3267 3269 3274 3276 3278 3280 3282 3284 3286 3288 3290 3295 3297 3299
              3301 3303 3305 3307 3309 3311 3417 3419 3421 3423 3425 3427 3429 3431
              3436 3438 3440 3442 3444 3446 3451 3453 3455 3457 3459 3461 3466 3468
              3470 3472 3474 3476 3478 3480 3482 3487 3489 3491 3493 3495 3497 3499
              3501 3503 3609 3611 3613 3615 3617 3619 3621 3623 3628 3630 3632 3634
              3636 3638 3643 3645 3647 3649 3651 3653 3658 3660 3662 3664 3666 3668
              3670 3672 3674 3679 3681 3683 3685 3687 3689 3691 3693 3695 3839 3841
              3843 3848 3850 3852 3857 3859 3861 3863 3865 3870 3872 3874 3876 3878
              3880 3882 3884 3886 3891 3893 3895 3897 3899 3901 3903 3905 3907 4043
              4045 4047 4052 4054 4056 4058 4063 4065 4067 4069 4074 4076 4078 4080
              4082 4084 4086 4088 4090 4095 4097 4099 4101 4103 4105 4107 4109 4111
              4235 4237 4239 4244 4246 4248 4253 4255 4257 4262 4264 4266 4271 4273
              4275 4277 4279 4281 4283 4285 4287 4292 4294 4296 4298 4300 4302 4304
              4306 4308 4472 4474 4476 4482 4484 4486 4488 4494 4496 4498 4500 4505
              4507 4509 4511 4513 4518 4520 4522 4524 4526 4531 4533 4535 4540 4542
              4544 4546 4548 4550 4552 4554 4556 4558 4560 4562 4564 4569 4571 4573
              4575 4577 4579 4581 4583 4585 4590 4592 4594 4596 4598 4600 4602 4604
              4606 4766 4768 4770 4776 4778 4780 4782 4784 4786 4788 4790 4792 4794

1596          4799 4801 4803 4805 4810 4812 4814 4816 4818 4820 4822 4824 4826 4831   SEQ 1609
```

```
Q 1596      4799  4801  4803  4805  4810  4812  4814  4816  4818  4820  4822  4824  4826  4831     SEQ 1609
            4833  4835  4837  4839  4841  4843  4845  4847  5048  5050  5052  5058  5060  5062
            5064  5066  5068  5070  5072  5079  5081  5083  5085  5087  5089  5091  5093  5098
            5100  5102  5104  5110  5112  5114  5116  5118  5124  5126  5128  5130  5132  5137
            5139  5141  5143  5145  5147  5149  5151  5153  5158  5160  5162  5164  5166  5168
            5170  5172  5174  5324  5326  5328  5330  5332  5337  5339  5341  5343  5345  5347
            5349  5351  5353  5355  5360  5362  5364  5366  5368  5370  5372  5374  5379  5381
            5383  5385  5387  5389  5391  5393  5398  5400  5402  5404  5406  5408  5413  5415
            5420  5422  5424  5426  5428  5430  5432  5434  5436  5441  5443  5445  5447  5449
            5451  5453  5455  5457  5595  5597  5599  5601  5606  5608  5610  5612  5614  5616
            5621  5623  5625  5627  5629  5631  5636  5638  5640  5642  5644  5646  5648  5653
            5655  5657  5659  5661  5663  5665  5667  5669  5674  5676  5678  5680  5682  5684
            5686  5688  5690  5811  5813  5815  5818  5820  5822  5941  5946  5948  5950  5952
            5954  5956  5958  5960  5962  5964  5966  5968  5970  5972  5974  5976  5978  5980
            5982  5984  5986  5988  5990  5992  5994  5996  5998  6000  6002  6004  6006  6008
            6010  6012  6014  6016  6018  6020  6022  6024  6026  6028  6030  6032  6034  6036
            6038  6040  6042  6044  6046  6048  6050  6052  6054  6056  6058  6060  6062  6064
            6066  6068  6070  6072  6074  6076  6078  6080  6082  6084  6086  6088  6090  6092
            6094  6096  6098  6100  6102  6104  6106  6108  6110  6112  6114  6116  6118  6120
            6122  6124  6126  6128  6130  6132  6134  6136  6138  6140  6142  6144  6146  6148
            6150  6152  6154  6156  6158  6160  6162  6164  6166  6168  6173  6181  6183  6185
            6187  6189  6194  6196  6198  6200  6202  6204  6206  6208  6210  6212  6214  6216
            6218  6220  6222  6224  6226  6228  6230  6232  6234  6236  6238  6240  6242  6244
            6246  6248  6250  6252  6254  6256  6258  6263  6265  6267  6269  6272  6274  6276
            6278  6281  6283  6285  6287  6290  6292  6294  6296  6299  6301  6303  6305  6308
            6310  6312  6314  6317  6319  6321  6323  6326  6328  6330  6332  6335  6337  6339
            6341  6344  6346  6348  6350  6353  6355  6357  6359  6362  6364  6366  6368  6371
            6373  6375  6377  6380  6382  6384  6386  6389  6391  6393  6395  6398  6400  6402
            6404  6407  6409  6411  6413  6416  6418  6420  6422  6425  6427  6429  6431  6434
            6436  6438  6440  6443  6445  6447  6449  6452  6454  6456  6458  6461  6463  6465
            6467  6470  6472  6474  6476  6479  6481  6483  6485  6488  6490  6492  6494  6497
            6499  6501  6503  6506  6508  6510  6512  6515  6517  6519  6521  6524  6526  6528
            6530  6533  6535  6537  6539  6542  6544  6546  6548  6551  6741  6743  6745  6747
            6753  6755  6841  6843  6845  6847  6852  6854  6859  6861  6863  6865  6867  6961
            6963  6965  6967  6969
   GET       673   922  6704  7178  7179
   GO         85    97   101   115   116   246   258   263   277   278   279   337   342   350
             351   353   355   368   369   427   431   433   444   446   447   465   467   468
             541   545   547   558   560   561   563   564   582   584   585   587   588   657
             658   672   688   690   695   723   727   906   907   921   937   939   944   972
             976  1158  1160  1166  1177  1276  1278  1284  1295  1368  1370  1376  1387  1512
            1514  1520  1531  1630  1632  1644  1645  1707  1710  1712  1713  1719  1721  1725
            1923  1925  1937  1938  1996  2000  2178  2180  2192  2193  2763  2765  2777  2778
            2826  2985  2987  2999  3000  3034  3162  3164  3176  3177  3354  3356  3368  3369
            3546  3548  3560  3561  3742  3744  3756  3757  3813  3816  3952  3954  3966  3967
            4020  4021  4157  4159  4171  4172  4357  4359  4365  4371  4373  4377  4378  4441
            4445  4651  4653  4665  4666  4716  4720  4738  4893  4895  4907  4908  5008  5009
            5010  5011  5012  5013  5014  5015  5016  5017  5018  5019  5020  5021  5022  5023
            5219  5221  5230  5235  5237  5241  5242  5295  5297  5301  5500  5502  5514  5515
            5570  5573  5731  5733  5739  5745  5794  5866  5868  5874  5880  5901  5903  5905
            5906  6589  6594  6600  6603  6636  6639  6643  6659  6664  6673  6675  6677  6679
            6688  6691  6700  6793  6795  6801  6807  6905  6907  6913  6919  7006  7019  7025
            7027  7031  7033  7035  7037  7039  7040  7051  7053  7057  7059  7061  7063  7065
```

```
            7066  7094  7096  7098  7137  7143  7191  7195  7210  7222  7226  7241
```

```
Q 1597        7066    7094    7096    7098    7137    7143    7191    7195    7210    7222    7226    7241
     MFLD     1213#   1213    1215#   1215    1217#   1217    1220#   1220    1222#   1222    1224#   1224    1227#   1227      SEQ 1610
              1229#   1229    1231#   1231    1234#   1234    1236#   1236    1238#   1238    1328#   1328    1330#   1330
              1332#   1332    1424#   1424    1426#   1426    1428#   1428    1431#   1431    1433#   1433    1435#   1435
              1438#   1438    1440#   1440    1442#   1442    1445#   1445    1447#   1447    1449#   1449    1452#   1452
              1454#   1454    1456#   1456    1459#   1459    1461#   1461    1463#   1463    1466#   1466    1468#   1468
              1470#   1470    1473#   1473    1475#   1475    1477#   1477    1564#   1564    1566#   1566    1568#   1568
              1571#   1571    1573#   1573    1575#   1575    1578#   1578    1580#   1580    1582#   1582    1749#   1749
              1751#   1751    1753#   1753    1758#   1758    1760#   1760    1762#   1762    1764#   1764    1769#   1769
              1771#   1771    1773#   1773    1775#   1775    1780#   1780    1782#   1782    1784#   1784    1786#   1786
              1788#   1788    1790#   1790    1792#   1792    1797#   1797    1799#   1799    1801#   1801    1803#   1803
              1805#   1805    1807#   1807    1809#   1809    1814#   1814    1816#   1816    1818#   1818    1820#   1820
              1822#   1822    1824#   1824    1826#   1826    1831#   1831    1833#   1833    1838#   1838    1840#   1840
              1842#   1842    1844#   1844    1846#   1846    1848#   1848    1850#   1850    1852#   1852    1854#   1854
              1859#   1859    1861#   1861    1863#   1863    1865#   1865    1867#   1867    1869#   1869    1871#   1871
              1873#   1873    1875#   1875    2026#   2026    2028#   2028    2030#   2030    2035#   2035    2037#   2037
              2039#   2039    2044#   2044    2046#   2046    2048#   2048    2050#   2050    2052#   2052    2054#   2054
              2059#   2059    2061#   2061    2063#   2063    2065#   2065    2067#   2067    2069#   2069    2071#   2071
              2077#   2077    2079#   2079    2081#   2081    2083#   2083    2085#   2085    2087#   2087    2089#   2089
              2091#   2091    2096#   2096    2098#   2098    2100#   2100    2102#   2102    2104#   2104    2106#   2106
              2108#   2108    2110#   2110    2112#   2112    2117#   2117    2119#   2119    2121#   2121    2123#   2123
              2125#   2125    2127#   2127    2129#   2129    2131#   2131    2133#   2133    2248#   2248    2250#   2250
              2252#   2252    2254#   2254    2256#   2256    2258#   2258    2260#   2260    2262#   2262    2264#   2264
              2266#   2266    2268#   2268    2270#   2270    2272#   2272    2274#   2274    2276#   2276    2278#   2278
              2280#   2280    2282#   2282    2284#   2284    2286#   2286    2288#   2288    2290#   2290    2292#   2292
              2294#   2294    2296#   2296    2298#   2298    2300#   2300    2302#   2302    2304#   2304    2306#   2306
              2308#   2308    2310#   2310    2312#   2312    2314#   2314    2316#   2316    2318#   2318    2320#   2320
              2322#   2322    2327#   2327    2329#   2329    2331#   2331    2336#   2336    2338#   2338    2340#   2340
              2342#   2342    2347#   2347    2349#   2349    2351#   2351    2356#   2356    2358#   2358    2360#   2360
              2365#   2365    2367#   2367    2369#   2369    2374#   2374    2376#   2376    2378#   2378    2383#   2383
              2385#   2385    2387#   2387    2392#   2392    2394#   2394    2396#   2396    2401#   2401    2403#   2403
              2405#   2405    2410#   2410    2412#   2412    2414#   2414    2419#   2419    2421#   2421    2423#   2423
              2428#   2428    2430#   2430    2432#   2432    2437#   2437    2439#   2439    2441#   2441    2446#   2446
              2448#   2448    2450#   2450    2455#   2455    2457#   2457    2459#   2459    2464#   2464    2466#   2466
              2468#   2468    2473#   2473    2475#   2475    2477#   2477    2482#   2482    2484#   2484    2486#   2486
              2491#   2491    2493#   2493    2495#   2495    2500#   2500    2502#   2502    2504#   2504    2509#   2509
              2511#   2511    2513#   2513    2518#   2518    2520#   2520    2522#   2522    2527#   2527    2529#   2529
              2531#   2531    2536#   2536    2538#   2538    2540#   2540    2545#   2545    2547#   2547    2549#   2549
              2554#   2554    2556#   2556    2558#   2558    2563#   2563    2565#   2565    2567#   2567    2572#   2572
              2574#   2574    2576#   2576    2581#   2581    2583#   2583    2585#   2585    2590#   2590    2592#   2592
              2594#   2594    2599#   2599    2601#   2601    2603#   2603    2608#   2608    2610#   2610    2612#   2612
              2617#   2617    2619#   2619    2621#   2621    2626#   2626    2628#   2628    2630#   2630    2635#   2635
              2637#   2637    2639#   2639    2644#   2644    2646#   2646    2648#   2648    2653#   2653    2655#   2655
              2657#   2657    2659#   2659    2664#   2664    2666#   2666    2671#   2671    2673#   2673    2675#   2675
              2680#   2680    2682#   2682    2684#   2684    2686#   2686    2688#   2688    2690#   2690    2692#   2692
              2694#   2694    2696#   2696    2701#   2701    2703#   2703    2705#   2705    2707#   2707    2709#   2709
              2711#   2711    2713#   2713    2715#   2715    2717#   2717    2848#   2848    2850#   2850    2852#   2852
              2857#   2857    2859#   2859    2861#   2861    2863#   2863    2865#   2865    2867#   2867    2872#   2872
              2874#   2874    2876#   2876    2878#   2878    2880#   2880    2882#   2882    2887#   2887    2889#   2889
              2891#   2891    2893#   2893    2895#   2895    2897#   2897    2899#   2899    2904#   2904    2906#   2906
              2908#   2908    2910#   2910    2912#   2912    2914#   2914    2916#   2916    2918#   2919    2920#   2920
              2925#   2925    2927#   2927    2929#   2929    2931#   2931    2933#   2933    2935#   2935    2937#   2937
              2939#   2939    2941#   2941    3054#   3054    3056#   3056    3058#   3058    3063#   3063    3065#   3065
```

```
              3067#   3067    3069#   3069    3071#   3071    3073#   3073    3075#   3075    3077#   3077    3082#   3082
```

```
EQ 1598   3067#  3067   3069#  3069   3071#  3071   3073#  3073   3075#  3075   3077#  3077   3082#  3082
          3084#  3084   3086#  3086   3088#  3088   3090#  3090   3092#  3092   3094#  3094   3096#  3096   SEQ 1611
          3098#  3098   3103#  3103   3105#  3105   3107#  3107   3109#  3109   3111#  3111   3113#  3113
          3115#  3115   3117#  3117   3119#  3119   3225#  3225   3227#  3227   3229#  3229   3231#  3231
          3233#  3233   3235#  3235   3237#  3237   3239#  3239   3244#  3244   3246#  3246   3248#  3248
          3250#  3250   3252#  3252   3254#  3254   3259#  3259   3261#  3261   3263#  3263   3265#  3265
          3267#  3267   3269#  3269   3274#  3274   3276#  3276   3278#  3278   3280#  3280   3282#  3282
          3284#  3284   3286#  3286   3288#  3288   3290#  3290   3295#  3295   3297#  3297   3299#  3299
          3301#  3301   3303#  3303   3305#  3305   3307#  3307   3309#  3309   3311#  3311   3417#  3417
          3419#  3419   3421#  3421   3423#  3423   3425#  3425   3427#  3427   3429#  3429   3431#  3431
          3436#  3436   3438#  3438   3440#  3440   3442#  3442   3444#  3444   3446#  3446   3451#  3451
          3453#  3453   3455#  3455   3457#  3457   3459#  3459   3461#  3461   3466#  3466   3468#  3468
          3470#  3470   3472#  3472   3474#  3474   3476#  3476   3478#  3478   3480#  3480   3482#  3482
          3487#  3487   3489#  3489   3491#  3491   3493#  3493   3495#  3495   3497#  3497   3499#  3499
          3501#  3501   3503#  3503   3609#  3609   3611#  3611   3613#  3613   3615#  3615   3617#  3617
          3619#  3619   3621#  3621   3623#  3623   3628#  3628   3630#  3630   3632#  3632   3634#  3634
          3636#  3636   3638#  3638   3643#  3643   3645#  3645   3647#  3647   3649#  3649   3651#  3651
          3653#  3653   3658#  3658   3660#  3660   3662#  3662   3664#  3664   3666#  3666   3668#  3668
          3670#  3670   3672#  3672   3674#  3674   3679#  3679   3681#  3681   3683#  3683   3685#  3685
          3687#  3687   3689#  3689   3691#  3691   3693#  3693   3695#  3695   3839#  3839   3841#  3841
          3843#  3843   3848#  3848   3850#  3850   3852#  3852   3857#  3857   3859#  3859   3861#  3861
          3863#  3863   3865#  3865   3870#  3870   3872#  3872   3874#  3874   3876#  3876   3878#  3878
          3880#  3880   3882#  3882   3884#  3884   3886#  3886   3891#  3891   3893#  3893   3895#  3895
          3897#  3897   3899#  3899   3901#  3901   3903#  3903   3905#  3905   3907#  3907   4043#  4043
          4045#  4045   4047#  4047   4052#  4052   4054#  4054   4056#  4056   4058#  4058   4063#  4063
          4065#  4065   4067#  4067   4069#  4069   4074#  4074   4076#  4076   4078#  4078   4080#  4080
          4082#  4082   4084#  4084   4086#  4086   4088#  4088   4090#  4090   4095#  4095   4097#  4097
          4099#  4099   4101#  4101   4103#  4103   4105#  4105   4107#  4107   4109#  4109   4111#  4111
          4235#  4235   4237#  4237   4239#  4239   4244#  4244   4246#  4246   4248#  4248   4253#  4253
          4255#  4255   4257#  4257   4262#  4262   4264#  4264   4266#  4266   4271#  4271   4273#  4273
          4275#  4275   4277#  4277   4279#  4279   4281#  4281   4283#  4283   4285#  4285   4287#  4287
          4292#  4292   4294#  4294   4296#  4296   4298#  4298   4300#  4300   4302#  4302   4304#  4304
          4306#  4306   4308#  4308   4472#  4472   4474#  4474   4476#  4476   4482#  4482   4484#  4484
          4486#  4486   4488#  4488   4494#  4494   4496#  4496   4498#  4498   4500#  4500   4505#  4505
          4507#  4507   4509#  4509   4511#  4511   4513#  4513   4518#  4518   4520#  4520   4522#  4522
          4524#  4524   4526#  4526   4531#  4531   4533#  4533   4535#  4535   4540#  4540   4542#  4542
          4544#  4544   4546#  4546   4548#  4548   4550#  4550   4552#  4552   4554#  4554   4556#  4556
          4558#  4558   4560#  4560   4562#  4562   4564#  4564   4569#  4569   4571#  4571   4573#  4573
          4575#  4575   4577#  4577   4579#  4579   4581#  4581   4583#  4583   4585#  4585   4590#  4590
          4592#  4592   4594#  4594   4596#  4596   4598#  4598   4600#  4600   4602#  4602   4604#  4604
          4606#  4606   4766#  4766   4768#  4768   4770#  4770   4776#  4776   4778#  4778   4780#  4780
          4782#  4782   4784#  4784   4786#  4786   4788#  4788   4790#  4790   4792#  4792   4794#  4794
          4799#  4799   4801#  4801   4803#  4803   4805#  4805   4810#  4810   4812#  4812   4814#  4814
          4816#  4816   4818#  4818   4820#  4820   4822#  4822   4824#  4824   4826#  4826   4831#  4831
          4833#  4833   4835#  4835   4837#  4837   4839#  4839   4841#  4841   4843#  4843   4845#  4845
          4847#  4847   5048#  5048   5050#  5050   5052#  5052   5058#  5058   5060#  5060   5062#  5062
          5064#  5064   5066#  5066   5068#  5068   5070#  5070   5072#  5072   5079#  5079   5081#  5081
          5083#  5083   5085#  5085   5087#  5087   5089#  5089   5091#  5091   5093#  5093   5098#  5098
          5100#  5100   5102#  5102   5104#  5104   5110#  5110   5112#  5112   5114#  5114   5116#  5116
          5118#  5118   5124#  5124   5126#  5126   5128#  5128   5130#  5130   5132#  5132   5137#  5137
          5139#  5139   5141#  5141   5143#  5143   5145#  5145   5147#  5147   5149#  5149   5151#  5151
          5153#  5153   5158#  5158   5160#  5160   5162#  5162   5164#  5164   5166#  5166   5168#  5168
          5170#  5170   5172#  5172   5174#  5174   5324#  5324   5326#  5326   5328#  5328   5330#  5330
```

```
          5332#  5332   5337#  5337   5339#  5339   5341#  5341   5343#  5343   5345#  5345   5347#  5347
```

EQ 1599

SEQ 1612

```
5332#  5332    5337#  5337    5339#  5339    5341#  5341    5343#  5343    5345#  5345    5347#  5347
5349#  5349    5351#  5351    5353#  5353    5355#  5355    5360#  5360    5362#  5362    5364#  5364
5366#  5366    5368#  5368    5370#  5370    5372#  5372    5374#  5374    5379#  5379    5381#  5381
5383#  5383    5385#  5385    5387#  5387    5389#  5389    5391#  5391    5393#  5393    5398#  5398
5400#  5400    5402#  5402    5404#  5404    5406#  5406    5408#  5408    5413#  5413    5415#  5415
5420#  5420    5422#  5422    5424#  5424    5426#  5426    5428#  5428    5430#  5430    5432#  5432
5434#  5434    5436#  5436    5441#  5441    5443#  5443    5445#  5445    5447#  5447    5449#  5449
5451#  5451    5453#  5453    5455#  5455    5457#  5457    5595#  5595    5597#  5597    5599#  5599
5601#  5601    5606#  5606    5608#  5608    5610#  5610    5612#  5612    5614#  5614    5616#  5616
5621#  5621    5623#  5623    5625#  5625    5627#  5627    5629#  5629    5631#  5631    5636#  5636
5638#  5638    5640#  5640    5642#  5642    5644#  5644    5646#  5646    5648#  5648    5653#  5653
5655#  5655    5657#  5657    5659#  5659    5661#  5661    5663#  5663    5665#  5665    5667#  5667
5669#  5669    5674#  5674    5676#  5676    5678#  5678    5680#  5680    5682#  5682    5684#  5684
5686#  5686    5688#  5688    5690#  5690    5811#  5811    5813#  5813    5815#  5815    5818#  5818
5820#  5820    5822#  5822    5941#  5941    5946#  5946    5948#  5948    5950#  5950    5952#  5952
5954#  5954    5956#  5956    5958#  5958    5960#  5960    5962#  5962    5964#  5964    5966#  5966
5968#  5968    5970#  5970    5972#  5972    5974#  5974    5976#  5976    5978#  5978    5980#  5980
5982#  5982    5984#  5984    5986#  5986    5988#  5988    5990#  5990    5992#  5992    5994#  5994
5996#  5996    5998#  5998    6000#  6000    6002#  6002    6004#  6004    6006#  6006    6008#  6008
6010#  6010    6012#  6012    6014#  6014    6016#  6016    6018#  6018    6020#  6020    6022#  6022
6024#  6024    6026#  6026    6028#  6028    6030#  6030    6032#  6032    6034#  6034    6036#  6036
6038#  6038    6040#  6040    6042#  6042    6044#  6044    6046#  6046    6048#  6048    6050#  6050
6052#  6052    6054#  6054    6056#  6056    6058#  6058    6060#  6060    6062#  6062    6064#  6064
6066#  6066    6068#  6068    6070#  6070    6072#  6072    6074#  6074    6076#  6076    6078#  6078
6080#  6080    6082#  6082    6084#  6084    6086#  6086    6088#  6088    6090#  6090    6092#  6092
6094#  6094    6096#  6096    6098#  6098    6100#  6100    6102#  6102    6104#  6104    6106#  6106
6108#  6108    6110#  6110    6112#  6112    6114#  6114    6116#  6116    6118#  6118    6120#  6120
6122#  6122    6124#  6124    6126#  6126    6128#  6128    6130#  6130    6132#  6132    6134#  6134
6136#  6136    6138#  6138    6140#  6140    6142#  6142    6144#  6144    6146#  6146    6148#  6148
6150#  6150    6152#  6152    6154#  6154    6156#  6156    6158#  6158    6160#  6160    6162#  6162
6164#  6164    6166#  6166    6168#  6168    6173#  6173    6181#  6181    6183#  6183    6185#  6185
6187#  6187    6189#  6189    6194#  6194    6196#  6196    6198#  6198    6200#  6200    6202#  6202
6204#  6204    6206#  6206    6208#  6208    6210#  6210    6212#  6212    6214#  6214    6216#  6216
6218#  6218    6220#  6220    6222#  6222    6224#  6224    6226#  6226    6228#  6228    6230#  6230
6232#  6232    6234#  6234    6236#  6236    6238#  6238    6240#  6240    6242#  6242    6244#  6244
6246#  6246    6248#  6248    6250#  6250    6252#  6252    6254#  6254    6256#  6256    6258#  6258
6263#  6263    6265#  6265    6267#  6267    6269#  6269    6272#  6272    6274#  6274    6276#  6276
6278#  6278    6281#  6281    6283#  6283    6285#  6285    6287#  6287    6290#  6290    6292#  6292
6294#  6294    6296#  6296    6299#  6299    6301#  6301    6303#  6303    6305#  6305    6308#  6308
6310#  6310    6312#  6312    6314#  6314    6317#  6317    6319#  6319    6321#  6321    6323#  6323
6326#  6326    6328#  6328    6330#  6330    6332#  6332    6335#  6335    6337#  6337    6339#  6339
6341#  6341    6344#  6344    6346#  6346    6348#  6348    6350#  6350    6353#  6353    6355#  6355
6357#  6357    6359#  6359    6362#  6362    6364#  6364    6366#  6366    6368#  6368    6371#  6371
6373#  6373    6375#  6375    6377#  6377    6380#  6380    6382#  6382    6384#  6384    6386#  6386
6389#  6389    6391#  6391    6393#  6393    6395#  6395    6398#  6398    6400#  6400    6402#  6402
6404#  6404    6407#  6407    6409#  6409    6411#  6411    6413#  6413    6416#  6416    6418#  6418
6420#  6420    6422#  6422    6425#  6425    6427#  6427    6429#  6429    6431#  6431    6434#  6434
6436#  6436    6438#  6438    6440#  6440    6443#  6443    6445#  6445    6447#  6447    6449#  6449
6452#  6452    6454#  6454    6456#  6456    6458#  6458    6461#  6461    6463#  6463    6465#  6465
6467#  6467    6470#  6470    6472#  6472    6474#  6474    6476#  6476    6479#  6479    6481#  6481
6483#  6483    6485#  6485    6488#  6488    6490#  6490    6492#  6492    6494#  6494    6497#  6497
6499#  6499    6501#  6501    6503#  6503    6506#  6506    6508#  6508    6510#  6510    6512#  6512
6515#  6515    6517#  6517    6519#  6519    6521#  6521    6524#  6524    6526#  6526    6528#  6528
```

```
              6530#  6530   6533#  6533   6535#  6535   6537#  6537   6539#  6539   6542#  6542   6544#  6544      SEQ 1613
              6546#  6546   6548#  6548   6551#  6551   6741#  6741   6743#  6743   6745#  6745   6747#  6747
              6753#  6753   6755#  6755   6841#  6841   6843#  6843   6845#  6845   6847#  6847   6852#  6852
              6854#  6854   6859#  6859   6861#  6861   6863#  6863   6865#  6865   6867#  6867   6961#  6961
              6963#  6963   6965#  6965   6967#  6967   6969#  6969
MWORD         1213   1215   1217   1220   1222   1224   1227   1229   1231   1234   1236   1238   1328   1330
              1332   1424   1426   1428   1431   1433   1435   1438   1440   1442   1445   1447   1449   1452
              1454   1456   1459   1461   1463   1466   1468   1470   1473   1475   1477   1564   1566   1568
              1571   1573   1575   1578   1580   1582   1749   1751   1753   1758   1760   1762   1764   1769
              1771   1773   1775   1780   1782   1784   1786   1788   1790   1792   1797   1799   1801   1803
              1805   1807   1809   1814   1816   1818   1820   1822   1824   1826   1831   1833   1838   1840
              1842   1844   1846   1848   1850   1852   1854   1859   1861   1863   1865   1867   1869   1871
              1873   1875   2026   2028   2030   2035   2037   2039   2044   2046   2048   2050   2052   2054
              2059   2061   2063   2065   2067   2069   2071   2077   2079   2081   2083   2085   2087   2089
              2091   2096   2098   2100   2102   2104   2106   2108   2110   2112   2117   2119   2121   2123
              2125   2127   2129   2131   2133   2248   2250   2252   2254   2256   2258   2260   2262   2264
              2266   2268   2270   2272   2274   2276   2278   2280   2282   2284   2286   2288   2290   2292
              2294   2296   2298   2300   2302   2304   2306   2308   2310   2312   2314   2316   2318   2320
              2322   2327   2329   2331   2336   2338   2340   2342   2347   2349   2351   2356   2358   2360
              2365   2367   2369   2374   2376   2378   2383   2385   2387   2392   2394   2396   2401   2403
              2405   2410   2412   2414   2419   2421   2423   2428   2430   2432   2437   2439   2441   2446
              2448   2450   2455   2457   2459   2464   2466   2468   2473   2475   2477   2482   2484   2486
              2491   2493   2495   2500   2502   2504   2509   2511   2513   2518   2520   2522   2527   2529
              2531   2536   2538   2540   2545   2547   2549   2554   2556   2558   2563   2565   2567   2572
              2574   2576   2581   2583   2585   2590   2592   2594   2599   2601   2603   2608   2610   2612
              2617   2619   2621   2626   2628   2630   2635   2637   2639   2644   2646   2648   2653   2655
              2657   2659   2664   2666   2671   2673   2675   2680   2682   2684   2686   2688   2690   2692
              2694   2696   2701   2703   2705   2707   2709   2711   2713   2715   2717   2848   2850   2852
              2857   2859   2861   2863   2865   2867   2872   2874   2876   2878   2880   2882   2887   2889
              2891   2893   2895   2897   2899   2904   2906   2908   2910   2912   2914   2916   2918   2920
              2925   2927   2929   2931   2933   2935   2937   2939   2941   3054   3056   3058   3063   3065
              3067   3069   3071   3073   3075   3077   3082   3084   3086   3088   3090   3092   3094   3096
              3098   3103   3105   3107   3109   3111   3113   3115   3117   3119   3225   3227   3229   3231
              3233   3235   3237   3239   3244   3246   3248   3250   3252   3254   3259   3261   3263   3265
              3267   3269   3274   3276   3278   3280   3282   3284   3286   3288   3290   3295   3297   3299
              3301   3303   3305   3307   3309   3311   3417   3419   3421   3423   3425   3427   3429   3431
              3436   3438   3440   3442   3444   3446   3451   3453   3455   3457   3459   3461   3466   3468
              3470   3472   3474   3476   3478   3480   3482   3487   3489   3491   3493   3495   3497   3499
              3501   3503   3609   3611   3613   3615   3617   3619   3621   3623   3628   3630   3632   3634
              3636   3638   3643   3645   3647   3649   3651   3653   3658   3660   3662   3664   3666   3668
              3670   3672   3674   3679   3681   3683   3685   3687   3689   3691   3693   3695   3839   3841
              3843   3848   3850   3852   3857   3859   3861   3863   3865   3870   3872   3874   3876   3878
              3880   3882   3884   3886   3891   3893   3895   3897   3899   3901   3903   3905   3907   4043
              4045   4047   4052   4054   4056   4058   4063   4065   4067   4069   4074   4076   4078   4080
              4082   4084   4086   4088   4090   4095   4097   4099   4101   4103   4105   4107   4109   4111
              4235   4237   4239   4244   4246   4248   4253   4255   4257   4262   4264   4266   4271   4273
              4275   4277   4279   4281   4283   4285   4287   4292   4294   4296   4298   4300   4302   4304
              4306   4308   4472   4474   4476   4482   4484   4486   4488   4494   4496   4498   4500   4505
              4507   4509   4511   4513   4518   4520   4522   4524   4526   4531   4533   4535   4540   4542
              4544   4546   4548   4550   4552   4554   4556   4558   4560   4562   4564   4569   4571   4573
              4575   4577   4579   4581   4583   4585   4590   4592   4594   4596   4598   4600   4602   4604
              4606   4766   4768   4770   4776   4778   4780   4782   4784   4786   4788   4790   4792   4794
              4799   4801   4803   4805   4810   4812   4814   4816   4818   4820   4822   4824   4826   4831
```

```
              4833   4835   4837   4839   4841   4843   4845   4847   5048   5050   5052   5058   5060   5062
              5064   5066   5068   5070   5072   5079   5081   5083   5085   5087   5089   5091   5093   5098      SEQ 1614
```

```
           4833   4835   4837   4839   4841   4843   4845   4847   5048   5050   5052   5058   5060   5062     SEQ 1614
           5064   5066   5068   5070   5072   5079   5081   5083   5085   5087   5089   5091   5093   5098
           5100   5102   5104   5110   5112   5114   5116   5118   5124   5126   5128   5130   5132   5137
           5139   5141   5143   5145   5147   5149   5151   5153   5158   5160   5162   5164   5166   5168
           5170   5172   5174   5324   5326   5328   5330   5332   5337   5339   5341   5343   5345   5347
           5349   5351   5353   5355   5360   5362   5364   5366   5368   5370   5372   5374   5379   5381
           5383   5385   5387   5389   5391   5393   5398   5400   5402   5404   5406   5408   5413   5415
           5420   5422   5424   5426   5428   5430   5432   5434   5436   5441   5443   5445   5447   5449
           5451   5453   5455   5457   5595   5597   5599   5601   5606   5608   5610   5612   5614   5616
           5621   5623   5625   5627   5629   5631   5636   5638   5640   5642   5644   5646   5648   5653
           5655   5657   5659   5661   5663   5665   5667   5669   5674   5676   5678   5680   5682   5684
           5686   5688   5690   5811   5813   5815   5818   5820   5822   5941   5946   5948   5950   5952
           5954   5956   5958   5960   5962   5964   5966   5968   5970   5972   5974   5976   5978   5980
           5982   5984   5986   5988   5990   5992   5994   5996   5998   6000   6002   6004   6006   6008
           6010   6012   6014   6016   6018   6020   6022   6024   6026   6028   6030   6032   6034   6036
           6038   6040   6042   6044   6046   6048   6050   6052   6054   6056   6058   6060   6062   6064
           6066   6068   6070   6072   6074   6076   6078   6080   6082   6084   6086   6088   6090   6092
           6094   6096   6098   6100   6102   6104   6106   6108   6110   6112   6114   6116   6118   6120
           6122   6124   6126   6128   6130   6132   6134   6136   6138   6140   6142   6144   6146   6148
           6150   6152   6154   6156   6158   6160   6162   6164   6166   6168   6173   6181   6183   6185
           6187   6189   6194   6196   6198   6200   6202   6204   6206   6208   6210   6212   6214   6216
           6218   6220   6222   6224   6226   6228   6230   6232   6234   6236   6238   6240   6242   6244
           6246   6248   6250   6252   6254   6256   6258   6263   6265   6267   6269   6272   6274   6276
           6278   6281   6283   6285   6287   6290   6292   6294   6296   6299   6301   6303   6305   6308
           6310   6312   6314   6317   6319   6321   6323   6326   6328   6330   6332   6335   6337   6339
           6341   6344   6346   6348   6350   6353   6355   6357   6359   6362   6364   6366   6368   6371
           6373   6375   6377   6380   6382   6384   6386   6389   6391   6393   6395   6398   6400   6402
           6404   6407   6409   6411   6413   6416   6418   6420   6422   6425   6427   6429   6431   6434
           6436   6438   6440   6443   6445   6447   6449   6452   6454   6456   6458   6461   6463   6465
           6467   6470   6472   6474   6476   6479   6481   6483   6485   6488   6490   6492   6494   6497
           6499   6501   6503   6506   6508   6510   6512   6515   6517   6519   6521   6524   6526   6528
           6530   6533   6535   6537   6539   6542   6544   6546   6548   6551   6741   6743   6745   6747
           6753   6755   6841   6843   6845   6847   6852   6854   6859   6861   6863   6865   6867   6961
           6963   6965   6967   6969
PNT3       5031   5040
PNT4        497    617    734    738    741    983    987    990   5921   6734   7126   7157   7163
PNTDEC      745    994
PNTHW      2242   5928   5934
PNTMSG      500    502    620    622    722    732    736    739    742    746    971    981    985    988
            991    995   1738   1739   1740   1741   1742   1743   2013   2014   2015   2016   2018   2020
           2238   2240   2243   2839   2840   2841   2842   3047   3048   3218   3219   3410   3411   3602
           3603   3829   3830   3831   3832   3833   4034   4035   4037   4226   4227   4228   4229   4458
           4460   4462   4464   4466   4733   4735   4736   4741   4743   4745   4747   4749   4751   4753
           4755   4757   4759   5007   5024   5026   5032   5035   5041   5314   5315   5316   5317   5586
           5587   5588   5589   5803   5806   5914   5916   5924   5926   5930   5932   6731   6732   6736
           7093   7123   7129   7131   7133   7135   7139   7141   7153   7155   7159   7161   7190   7200
           7201   7203   7205   7206   7221   7231   7232   7234   7236   7237
PNTOCS     5805   7176
PUT         670    919   6646   7170   7171
RGET       7178
RPUT       7170
RTN         132    295    383    485    503    605    623    712    728    735    747    961    977    984
            996   1161   1197   1279   1315   1371   1405   1515   1549   1633   1659   1714   1726   1736
```

```
           1926   1952   1997   2001   2011   2181   2207   2224   2229   2239   2244   2766   2792   2827     SEQ 1615
           2837   2988   3014   3035   3045   3165   3191   3216   3357   3383   3408   3549   3575   3600
```

```
          1926  1952  1997  2001  2011  2181  2207  2224  2229  2239  2244  2766  2792  2827
          2837  2988  3014  3035  3045  3165  3191  3216  3357  3383  3408  3549  3575  3600
          3745  3771  3814  3817  3827  3955  3981  4011  4017  4022  4032  4160  4186  4224
          4360  4392  4446  4456  4654  4680  4721  4731  4740  4760  4896  4922  5005  5033
          5042  5222  5256  5302  5312  5503  5529  5574  5584  5734  5759  5775  5796  5807
          5869  5891  5935  6607  6618  6705  6722  6737  6796  6821  6908  6933  6946  6952
          7083  7099  7114  7144  7164  7179  7196  7211  7227  7242
SCOPER     120   283   373   451   472   568   592   699   948  1184  1302  1392  1536  1652
          1945  2200  2785  3007  3184  3376  3568  3764  3974  4179  4385  4673  4915  5249
          5522  5752  5885  6611  6814  6926  7044  7070
STABLE    6937  6938  6939  6940  6941  6942
TMSG       500   502   620   622   742   746   991   995  1733  1739  1740  1741  1742  1743
          2013  2014  2015  2016  2017  2019  2238  2240  2243  2839  2840  2841  2842  3047
          3048  3218  3219  3410  3411  3602  3603  3829  3830  3831  3832  3833  4034  4035
          4036  4226  4227  4228  4229  4458  4459  4461  4463  4465  4733  4734  4736  4743
          4745  4747  4749  4751  4753  4755  4757  4759  5007  5024  5026  5035  5314  5315
          5316  5317  5586  5587  5588  5589  5924  6736  7129  7131
TMSGC      722   732   736   739   971   981   985   988  4741  5032  5041  5803  5806  5914
          5916  5926  5930  5932  6731  6732  7093  7123  7133  7135  7139  7141  7153  7155
          7159  7161  7190  7200  7203  7206  7221  7231  7234  7237
TTABLE    1665  1666  1667  1668  1672  1673  1674  1675  1679  1680  1681  1682  1687  1688
          1689  1690  1694  1695  1696  1697  1701  1702  1703  1704  1705  1958  1959  1960
          1964  1965  1966  1970  1971  1972  1977  1978  1979  1983  1984  1985  1986  1990
          1991  1992  1993  1994  2211  2212  2213  2214  2215  2798  2799  2800  2804  2805
          2806  2810  2811  2812  2813  2818  2819  2821  3020  3021  3022  3026  3027  3028
          3029  3197  3198  3199  3203  3204  3205  3206  3389  3390  3391  3395  3396  3397
          3398  3581  3582  3583  3587  3588  3589  3590  3777  3778  3779  3783  3784  3785
          3786  3790  3791  3792  3793  3798  3799  3800  3801  3805  3806  3807  3808  3809
          3987  3988  3989  3993  3994  3995  3996  3997  4001  4002  4003  4004  4005  4192
          4193  4194  4198  4199  4200  4204  4205  4206  4211  4212  4213  4214  4398  4399
          4400  4405  4406  4407  4408  4414  4415  4416  4417  4421  4422  4423  4424  4428
          4429  4430  4431  4432  4686  4687  4688  4692  4693  4694  4695  4699  4700  4701
          4928  4929  4930  4935  4936  4937  4938  4939  4940  4941  4942  4943  4944  4945
          4946  4947  4948  4949  4951  4952  4953  4954  4955  4956  4957  4958  4959  4965
          4966  4967  4968  4969  4970  4971  4972  4973  4974  4975  4976  4977  4978  4979
          4980  4981  4982  4983  4984  4985  4986  4987  4988  4992  4993  4994  4995  5262
          5263  5264  5265  5269  5270  5271  5272  5277  5278  5279  5280  5284  5285  5286
          5287  5288  5535  5536  5537  5538  5542  5543  5544  5545  5549  5550  5551  5552
          5556  5557  5558  5559  5560
```

```
                                        TITLE   DFPTA9
                                        SUBTTL  CBUS MODULE TESTS

                                        SALL

                                        SEARCH  DFPTA,MONSYM

                                ; INTERN's

                                        ENTRY   TSTC1,TSTC2,TSTC3,TSTC4,TSTC5,TSTC6,TSTC7,TSTC10
                                        ENTRY   TSTC11,TSTC12,TSTC13,TSTC14,TSTC15,TSTC16,TSTC17,TSTC20
                                        ENTRY   TSTC21,TSTC22,TSTC23,TSTC24,TSTC25,TSTC26,TSTC27,TSTC30
                                        ENTRY   TSTC31,TSTC32,TSTC33,TSTC34,TSTC35,TSTC36,TSTC37

                                ; EXTERN's

                                        EXTERN  TSTC40,TSTC41

                                ; EXTERN's located in DFPTA1.MAC (Mainline code..)

                                        EXTERN  UDEBUG,BUFF,CBUF,TLOAD,CSRPNT,TRACE,TSTSUB,ODELAY,TSLOD1,TSLOD2
                                        EXTERN  BEXEC,BBPNT,TEXEC,TTPNT,TEBUFC,TEBUFA,UDEBUG,PORTCI,PORTNI
                                        EXTERN  CEXEC,CCPNT,CPAT,CSTATF,CEBUFA,CDERR,CLEN

                                ; EXTERN's located in DFPTAI.MAC (Port Device Handling Routines module)

                                        EXTERN  LDRAR,LDCSR,RDEBUF,RDCSR,IPACLR,IPASRT,SNEXT,SDATA
                                        EXTERN  CHINIT,GENCCW

                                ; EXTERN's located in DFPTAU.MAC (Utility Routines module)

                                        EXTERN  PATBUF,BUFCOM,SCOSW,BUFGEN


                                ;#*********************************************************************
                                ; Z9 - Address for use in DDT
                                ;#*********************************************************************

        000000'                 Z9:                                     ; address of 00000'
```

```
                                        ;#*****************************************************************
                                        ;* Test 1 - Fmtr Data Loopback
                                        ;
                                        ; Description:  This test does simple data loopback over the MBUS
                                        ;               through the CBUF, through the Fmtr,  through  the
                                        ;               CMUX and back to the MBUS.
                                        ;
                                        ; Procedure:    Port Clear
                                        ;               Set RAR to 0
                                        ;               Execute single steps
                                        ;               Read EBUF, verify data is correct
                                        ;
                                        ;               Repeat with each data pattern:
                                        ;                 0's
                                        ;                 1's
                                        ;                 floating 1's
                                        ;                 floating 0's
                                        ;
                                        ; Failure:      ---
                                        ;#*****************************************************************

                                        ; Test data

 000000' 254 00 0 00 000011'            TSTC1:  JRST    TG1                     ; go start test
 000001' 110604  000001                         CBUS!MBUS!NDMP!NDCB!ZCBUS!1     ; test mask
 000002' 000152' 012703'                        T1M,,[ASCIZ ^Fmtr Data Loopback^]
 000003' 012707' 012713'                        [EXP C20,C18,C23,MLAST!C24],,[EXP C12,C8,C3,C4,MLAST!C2]
 000004' 000000  000637'                        TSTC2                           ; failure test table
 000005' 000000  003034'                        TSTC11                          ;   ...
 000006' 000000  001044'                        TSTC3
 000007' 000000  001152'                        TSTC4
 000010' 777777  777777                         -1

                                        ; Start test

 000011' 201 00 0 00 000000'            TG1:    MOVEI   Z9                      ; get address of module start
 000012' 260 17 0 00 000000*                    GO      TRACE                   ; handle trace output
 000013' 201 01 0 00 000152'                    MOVEI   1,T1M                   ; set up microcode address
 000014' 260 17 0 00 000000*                    GO      TLOAD                   ; load/verify it
 000015' 263 17 0 00 000000                     RTN                             ; failed - exit test

                                        ; Initialization

 000016' 400 15 0 00 000000             TL1:    SETZ    ERFLG,                  ; clear error flag
 000017' 260 17 0 00 000000*                    GO      IPACLR                  ; clear port
 000020' 402 00 0 00 000000*                    SETZM   TSTSUB                  ; initialize subtest number
 000021' 201 06 0 00 000033'                    MOVEI   6,TS1                   ; get sstep table address
```

```

                                          ; Loop on single step table entries

000022' 260 17 0 00 000000*               TA1:    GO      BEXEC                   ; execute table entry
000023' 254 00 0 00 000032'                       JRST    TX1                     ; end of sstep table
000024' 254 00 0 00 000022'                       JRST    TA1                     ; keep looping after call
000025' 474 15 0 00 000000                        SETO    ERFLG,                  ; error - EBUF data incorrect

                                          ; Handle error printouts and scope looping

000026' 027 00 0 00 000150'                       SCOPER  MA1                     ; print error message
000027' 254 00 0 00 000016'                       JRST    TL1                     ; loop on error
000030' 254 00 0 00 000032'                       JRST    TX1                     ; altmode exit
000031' 322 15 0 00 000022'                       JUMPE   ERFLG,TA1               ; do next sstep table entry

                                          ; End of test

000032' 263 17 0 00 000000                TX1:    RTN                             ; return

                                          ; Sstep table, with EBUF data, as:  (CMD,#Steps,Start addr,End Addr,EBUF data)

000033' 100600  005004                    TS1:    ATABLE  (SSSTRT,6,0,5004,-1)
000034' 777777  777777
000035' 100600  015004                            ATABLE  (SSSTRT,6,1,5004,1)
000036' 000000  000001
000037' 101000  025004                            ATABLE  (SSSTRT,^D8,2,5004,2)
000040' 000000  000002
000041' 101100  035004                            ATABLE  (SSSTRT,^D9,3,5004,4)
000042' 000000  000004
000043' 101200  045004                            ATABLE  (SSSTRT,^D10,4,5004,10)
000044' 000000  000010
000045' 101300  055004                            ATABLE  (SSSTRT,^D11,5,5004,20)
000046' 000000  000020
000047' 101400  065004                            ATABLE  (SSSTRT,^D12,6,5004,40)
000050' 000000  000040
000051' 101500  075004                            ATABLE  (SSSTRT,^D13,7,5004,100)
000052' 000000  000100
000053' 101600  105004                            ATABLE  (SSSTRT,^D14,10,5004,200)
000054' 000000  000200
000055' 101700  115004                            ATABLE  (SSSTRT,^D15,11,5004,400)
000056' 000000  000400
000057' 102000  125004                            ATABLE  (SSSTRT,^D16,12,5004,1000)
000060' 000000  001000
000061' 102100  135004                            ATABLE  (SSSTRT,^D17,13,5004,2000)
000062' 000000  002000
000063' 102200  145004                            ATABLE  (SSSTRT,^D18,14,5004,4000)
000064' 000000  004000
000065' 102300  155004                            ATABLE  (SSSTRT,^D19,15,5004,10000)
000066' 000000  010000
000067' 102400  165004                            ATABLE  (SSSTRT,^D20,16,5004,20000)
000070' 000000  020000
000071' 102500  175004                            ATABLE  (SSSTRT,^D21,17,5004,40000)
000072' 000000  040000
000073' 102600  205004                            ATABLE  (SSSTRT,^D22,20,5004,100000)
000074' 000000  100000
```

```
000075' 102700  215004                 ATABLE  (SSSTRT,^D23,21,5004,200000)
000076' 000000  200000
000077' 103000  225004                 ATABLE  (SSSTRT,^D24,22,5004,400000)
000100' 000000  400000
000101' 103100  235004                 ATABLE  (SSSTRT,^D25,23,5004,1000000)
000102' 000001  000000
000103' 103200  245004                 ATABLE  (SSSTRT,^D26,24,5004,2000000)
000104' 000002  000000
000105' 103300  255004                 ATABLE  (SSSTRT,^D27,25,5004,4000000)
000106' 000004  000000
000107' 103400  265004                 ATABLE  (SSSTRT,^D28,26,5004,10000000)
000110' 000010  000000
000111' 103500  275004                 ATABLE  (SSSTRT,^D29,27,5004,20000000)
000112' 000020  000000
000113' 103600  305004                 ATABLE  (SSSTRT,^D30,30,5004,40000000)
000114' 000040  000000
000115' 103700  315004                 ATABLE  (SSSTRT,^D31,31,5004,100000000)
000116' 000100  000000
000117' 104000  325004                 ATABLE  (SSSTRT,^D32,32,5004,200000000)
000120' 000200  000000
000121' 104100  335004                 ATABLE  (SSSTRT,^D33,33,5004,400000000)
000122' 000400  000000
000123' 104200  345004                 ATABLE  (SSSTRT,^D34,34,5004,1000000000)
000124' 001000  000000
000125' 104300  355004                 ATABLE  (SSSTRT,^D35,35,5004,2000000000)
000126' 002000  000000
000127' 104400  365004                 ATABLE  (SSSTRT,^D36,36,5004,4000000000)
000130' 004000  000000
000131' 104500  375004                 ATABLE  (SSSTRT,^D37,37,5004,10000000000)
000132' 010000  000000
000133' 104600  405004                 ATABLE  (SSSTRT,^D38,40,5004,20000000000)
000134' 020000  000000
000135' 104700  415004                 ATABLE  (SSSTRT,^D39,41,5004,40000000000)
000136' 040000  000000
000137' 105000  425004                 ATABLE  (SSSTRT,^D40,42,5004,100000000000)
000140' 100000  000000
000141' 105100  435004                 ATABLE  (SSSTRT,^D41,43,5004,200000000000)
000142' 200000  000000
000143' 105000  445004                 ATABLE  (SSSTRT,^D40,44,5004,400000000000)
000144' 400000  000000
000145' 100500  455004                 ATABLE  (SSSTRT,5,45,5004,0)
000146' 000000  000000
000147' 000000  000000                 ATABLE  (SSLAST)

                                ; Error messages

000150' 140000  012720'         MA1:    MSG!TXNOT![ASCIZ /Data loopback through Fmtr failed (result in EBUF)/]
000151' 000000000000#                   LAST!CALL!TXALL!BBPNT           ; print sequencer data

                                ; Microcode

000152' 000001  000000          T1M:    MWORD   <ADDR=0,JMAP,J=100>             ; 0 - 1's
000153' 000000  000040
000154' 000102  000000                  MWORD   <JMAP,J=200>                    ; 1 - 1
000155' 000000  000040
```

```
198  000156' 000203  000000          MWORD   <JMAP,J=300>              ; 2 - 2
199  000157' 000000  000040
200  000160' 000304  000000          MWORD   <JMAP,J=400>              ; 3 - 4
201  000161' 000000  000040
202  000162' 000405  000000          MWORD   <JMAP,J=500>              ; 4 - 10
203  000163' 000000  000040
204  000164' 000506  000000          MWORD   <JMAP,J=600>              ; 5 - 20
205  000165' 000000  000040
206  000166' 000607  000000          MWORD   <JMAP,J=700>              ; 6 - 40
207  000167' 000000  000040
208  000170' 000710  000000          MWORD   <JMAP,J=1000>             ; 7 - 100
209  000171' 000000  000040
210  000172' 001011  000000          MWORD   <JMAP,J=1100>             ; 10 - 200
211  000173' 000000  000040
212  000174' 001112  000000          MWORD   <JMAP,J=1200>             ; 11 - 400
213  000175' 000000  000040
214  000176' 001213  000000          MWORD   <JMAP,J=1300>             ; 12 - 1000
215  000177' 000000  000040
216  000200' 001314  000000          MWORD   <JMAP,J=1400>             ; 13 - 2000
217  000201' 000000  000040
218  000202' 001415  000000          MWORD   <JMAP,J=1500>             ; 14 - 4000
219  000203' 000000  000040
220  000204' 001516  000000          MWORD   <JMAP,J=1600>             ; 15 - 10000
221  000205' 000000  000040
222  000206' 001617  000000          MWORD   <JMAP,J=1700>             ; 16 - 20000
223  000207' 000000  000040
224  000210' 001720  000000          MWORD   <JMAP,J=2000>             ; 17 - 40000
225  000211' 000000  000040
226  000212' 002021  000000          MWORD   <JMAP,J=2100>             ; 20 - 100000
227  000213' 000000  000040
228  000214' 002122  000000          MWORD   <JMAP,J=2200>             ; 21 - 200000
229  000215' 000000  000040
230  000216' 002223  000000          MWORD   <JMAP,J=2300>             ; 22 - 400000
231  000217' 000000  000040
232  000220' 002324  000000          MWORD   <JMAP,J=2400>             ; 23 - 1,,0
233  000221' 000000  000040
234  000222' 002425  000000          MWORD   <JMAP,J=2500>             ; 24 - 2,,0
235  000223' 000000  000040
236  000224' 002526  000000          MWORD   <JMAP,J=2600>             ; 25 - 4,,0
237  000225' 000000  000040
238  000226' 002627  000000          MWORD   <JMAP,J=2700>             ; 26 - 10,,0
239  000227' 000000  000040
240  000230' 002730  000000          MWORD   <JMAP,J=3000>             ; 27 - 20,,0
241  000231' 000000  000040
242  000232' 003031  000000          MWORD   <JMAP,J=3100>             ; 30 - 40,,0
243  000233' 000000  000040
244  000234' 003132  000000          MWORD   <JMAP,J=3200>             ; 31 - 100,,0
245  000235' 000000  000040
246  000236' 003233  000000          MWORD   <JMAP,J=3300>             ; 32 - 200,,0
247  000237' 000000  000040
248  000240' 003334  000000          MWORD   <JMAP,J=3400>             ; 33 - 400,,0
249  000241' 000000  000040
250  000242' 003435  000000          MWORD   <JMAP,J=3500>             ; 34 - 1000,,0
251  000243' 000000  000040
252  000244' 003536  000000          MWORD   <JMAP,J=3600>             ; 35 - 2000,,0
```

```
000245' 000000  000040
000246' 003637  000000              MWORD   <JMAP,J=3700>                       ; 36 - 4000,,0
000247' 000000  000040
000250' 003740  000000              MWORD   <JMAP,J=4000>                       ; 37 - 10000,,0
000251' 000000  000040
000252' 004041  000000              MWORD   <JMAP,J=4100>                       ; 40 - 20000,,0
000253' 000000  000040
000254' 004142  000000              MWORD   <JMAP,J=4200>                       ; 41 - 40000,,0
000255' 000000  000040
000256' 004243  000000              MWORD   <JMAP,J=4300>                       ; 42 - 100000,,0
000257' 000000  000040
000260' 004344  000000              MWORD   <JMAP,J=4400>                       ; 43 - 200000,,0
000261' 000000  000040
000262' 004445  000000              MWORD   <JMAP,J=4500>                       ; 44 - 400000,,0
000263' 000000  000040
000264' 004546  000000              MWORD   <JMAP,J=4600>                       ; 45 - 0's
000265' 000000  000040

                            ; Data pattern 1's

000266' 010000  000000              MWORD   <ADDR=100,CONT,SDO,B=2,AND,D=2>             ; 100
000267' 742001  000340
000270' 010150  010000              MWORD   <JMAP,J=5001,SAB,A=2,B=2,XNOR,D=2>          ; 101
000271' 172021  000040

                            ; Data pattern floating 1's - 1

000272' 020000  000000              MWORD   <ADDR=200,CONT,SDO,B=2,AND,D=2>             ; 200
000273' 742001  000340
000274' 020150  010000              MWORD   <JMAP,J=5001,SAB,A=2,B=2,PLUS,CRY,D=2>      ; 201
000275' 102021  000440

                            ; Data pattern floating 1's - 2

000276' 030000  000000              MWORD   <ADDR=300,CONT,SDO,B=2,AND,D=2>             ; 300
000277' 742001  000340
000300' 030100  000000              MWORD   <LDCT,J=0,SAB,A=2,B=2,PLUS,CRY,D=2>         ; 301
000301' 102021  000700
000302' 030250  000000              MWORD   <JMAP,J=5000>                               ; 302
000303' 000000  000040

                            ; Data pattern floating 1's - 4

000304' 040000  000000              MWORD   <ADDR=400,CONT,SDO,B=2,AND,D=2>             ; 400
000305' 742001  000340
000306' 040100  010000              MWORD   <LDCT,J=1,SAB,A=2,B=2,PLUS,CRY,D=2>         ; 401
000307' 102021  000700
000310' 040250  000000              MWORD   <JMAP,J=5000>                               ; 402
000311' 000000  000040

                            ; Data pattern floating 1's - 10

000312' 050000  000000              MWORD   <ADDR=500,CONT,SDO,B=2,AND,D=2>             ; 500
000313' 742001  000340
000314' 050100  020000              MWORD   <LDCT,J=2,SAB,A=2,B=2,PLUS,CRY,D=2>         ; 501
```

```
000315' 102021  000700
000316' 050250  000000                  MWORD   <JMAP,J=5000>                           ; 502
000317' 000000  000040

                                ; Data pattern floating 1's - 20

000320' 060000  000000                  MWORD   <ADDR=600,CONT,SDO,B=2,AND,D=2>         ; 600
000321' 742001  000340
000322' 060100  030000                  MWORD   <LDCT,J=3,SAB,A=2,B=2,PLUS,CRY,D=2>     ; 601
000323' 102021  000700
000324' 060250  000000                  MWORD   <JMAP,J=5000>                           ; 602
000325' 000000  000040

                                ; Data pattern floating 1's - 40

000326' 070000  000000                  MWORD   <ADDR=700,CONT,SDO,B=2,AND,D=2>         ; 700
000327' 742001  000340
000330' 070100  040000                  MWORD   <LDCT,J=4,SAB,A=2,B=2,PLUS,CRY,D=2>     ; 701
000331' 102021  000700
000332' 070250  000000                  MWORD   <JMAP,J=5000>                           ; 702
000333' 000000  000040

                                ; Data pattern floating 1's - 100

000334' 100000  000000                  MWORD   <ADDR=1000,CONT,SDO,B=2,AND,D=2>        ; 1000
000335' 742001  000340
000336' 100100  050000                  MWORD   <LDCT,J=5,SAB,A=2,B=2,PLUS,CRY,D=2>     ; 1001
000337' 102021  000700
000340' 100250  000000                  MWORD   <JMAP,J=5000>                           ; 1002
000341' 000000  000040

                                ; Data pattern floating 1's - 200

000342' 110000  000000                  MWORD   <ADDR=1100,CONT,SDO,B=2,AND,D=2>        ; 1100
000343' 742001  000340
000344' 110100  060000                  MWORD   <LDCT,J=6,SAB,A=2,B=2,PLUS,CRY,D=2>     ; 1101
000345' 102021  000700
000346' 110250  000000                  MWORD   <JMAP,J=5000>                           ; 1102
000347' 000000  000040

                                ; Data pattern floating 1's - 400

000350' 120000  000000                  MWORD   <ADDR=1200,CONT,SDO,B=2,AND,D=2>        ; 1200
000351' 742001  000340
000352' 120100  070000                  MWORD   <LDCT,J=7,SAB,A=2,B=2,PLUS,CRY,D=2>     ; 1201
000353' 102021  000700
000354' 120250  000000                  MWORD   <JMAP,J=5000>                           ; 1202
000355' 000000  000040

                                ; Data pattern floating 1's - 1000

000356' 130000  000000                  MWORD   <ADDR=1300,CONT,SDO,B=2,AND,D=2>        ; 1300
000357' 742001  000340
000360' 130100  100000                  MWORD   <LDCT,J=10,SAB,A=2,B=2,PLUS,CRY,D=2>    ; 1301
000361' 102021  000700
```

```
000362' 130250  000000             MWORD   <JMAP,J=5000>                              ; 1302
000363' 000000  000040

                           ; Data pattern floating 1's - 2000

000364' 140000  000000             MWORD   <ADDR=1400,CONT,SDO,B=2,AND,D=2>           ; 1400
000365' 742001  000340
000366' 140100  110000             MWORD   <LDCT,J=11,SAB,A=2,B=2,PLUS,CRY,D=2>       ; 1401
000367' 102021  000700
000370' 140250  000000             MWORD   <JMAP,J=5000>                              ; 1402
000371' 000000  000040

                           ; Data pattern floating 1's - 4000

000372' 150000  000000             MWORD   <ADDR=1500,CONT,SDO,B=2,AND,D=2>           ; 1500
000373' 742001  000340
000374' 150100  120000             MWORD   <LDCT,J=12,SAB,A=2,B=2,PLUS,CRY,D=2>       ; 1501
000375' 102021  000700
000376' 150250  000000             MWORD   <JMAP,J=5000>                              ; 1502
000377' 000000  000040

                           ; Data pattern floating 1's - 10000

000400' 160000  000000             MWORD   <ADDR=1600,CONT,SDO,B=2,AND,D=2>           ; 1600
000401' 742001  000340
000402' 160100  130000             MWORD   <LDCT,J=13,SAB,A=2,B=2,PLUS,CRY,D=2>       ; 1601
000403' 102021  000700
000404' 160250  000000             MWORD   <JMAP,J=5000>                              ; 1602
000405' 000000  000040

                           ; Data pattern floating 1's - 20000

000406' 170000  000000             MWORD   <ADDR=1700,CONT,SDO,B=2,AND,D=2>           ; 1700
000407' 742001  000340
000410' 170100  140000             MWORD   <LDCT,J=14,SAB,A=2,B=2,PLUS,CRY,D=2>       ; 1701
000411' 102021  000700
000412' 170250  000000             MWORD   <JMAP,J=5000>                              ; 1702
000413' 000000  000040

                           ; Data pattern floating 1's - 40000

000414' 200000  000000             MWORD   <ADDR=2000,CONT,SDO,B=2,AND,D=2>           ; 2000
000415' 742001  000340
000416' 200100  150000             MWORD   <LDCT,J=15,SAB,A=2,B=2,PLUS,CRY,D=2>       ; 2001
000417' 102021  000700
000420' 200250  000000             MWORD   <JMAP,J=5000>                              ; 2002
000421' 000000  000040

                           ; Data pattern floating 1's - 100000

000422' 210000  000000             MWORD   <ADDR=2100,CONT,SDO,B=2,AND,D=2>           ; 2100
000423' 742001  000340
000424' 210100  160000             MWORD   <LDCT,J=16,SAB,A=2,B=2,PLUS,CRY,D=2>       ; 2101
000425' 102021  000700
000426' 210250  000000             MWORD   <JMAP,J=5000>                              ; 2102
```

```
000427' 000000  000040

                                ; Data pattern floating 1's - 200000

000430' 220000  000000                  MWORD   <ADDR=2200,CONT,SDO,B=2,AND,D=2>                ; 2200
000431' 742001  000340
000432' 220100  170000                  MWORD   <LDCT,J=17,SAB,A=2,B=2,PLUS,CRY,D=2>            ; 2201
000433' 102021  000700
000434' 220250  000000                  MWORD   <JMAP,J=5000>                                   ; 2202
000435' 000000  000040

                                ; Data pattern floating 1's - 400000

000436' 230000  000000                  MWORD   <ADDR=2300,CONT,SDO,B=2,AND,D=2>                ; 2300
000437' 742001  000340
000440' 230100  200000                  MWORD   <LDCT,J=20,SAB,A=2,B=2,PLUS,CRY,D=2>            ; 2301
000441' 102021  000700
000442' 230250  000000                  MWORD   <JMAP,J=5000>                                   ; 2302
000443' 000000  000040

                                ; Data pattern floating 1's - 1,,000000

000444' 240000  000000                  MWORD   <ADDR=2400,CONT,SDO,B=2,AND,D=2>                ; 2400
000445' 742001  000340
000446' 240100  210000                  MWORD   <LDCT,J=21,SAB,A=2,B=2,PLUS,CRY,D=2>            ; 2401
000447' 102021  000700
000450' 240250  000000                  MWORD   <JMAP,J=5000>                                   ; 2402
000451' 000000  000040

                                ; Data pattern floating 1's - 2,,000000

000452' 250000  000000                  MWORD   <ADDR=2500,CONT,SDO,B=2,AND,D=2>                ; 2500
000453' 742001  000340
000454' 250100  220000                  MWORD   <LDCT,J=22,SAB,A=2,B=2,PLUS,CRY,D=2>            ; 2501
000455' 102021  000700
000456' 250250  000000                  MWORD   <JMAP,J=5000>                                   ; 2502
000457' 000000  000040

                                ; Data pattern floating 1's - 4,,000000

000460' 260000  000000                  MWORD   <ADDR=2600,CONT,SDO,B=2,AND,D=2>                ; 2600
000461' 742001  000340
000462' 260100  230000                  MWORD   <LDCT,J=23,SAB,A=2,B=2,PLUS,CRY,D=2>            ; 2601
000463' 102021  000700
000464' 260250  000000                  MWORD   <JMAP,J=5000>                                   ; 2602
000465' 000000  000040

                                ; Data pattern floating 1's - 10,,000000

000466' 270000  000000                  MWORD   <ADDR=2700,CONT,SDO,B=2,AND,D=2>                ; 2700
000467' 742001  000340
000470' 270100  240000                  MWORD   <LDCT,J=24,SAB,A=2,B=2,PLUS,CRY,D=2>            ; 2701
000471' 102021  000700
000472' 270250  000000                  MWORD   <JMAP,J=5000>                                   ; 2702
000473' 000000  000040
```

```

                                ; Data pattern floating 1's - 20,,000000

000474' 300000  000000                  MWORD   <ADDR=3000,CONT,SDO,B=2,AND,D=2>            ; 3000
000475' 742001  000340
000476' 300100  250000                  MWORD   <LDCT,J=25,SAB,A=2,B=2,PLUS,CRY,D=2>        ; 3001
000477' 102021  000700
000500' 300250  000000                  MWORD   <JMAP,J=5000>                               ; 3002
000501' 000000  000040

                                ; Data pattern floating 1's - 40,,000000

000502' 310000  000000                  MWORD   <ADDR=3100,CONT,SDO,B=2,AND,D=2>            ; 3100
000503' 742001  000340
000504' 310100  260000                  MWORD   <LDCT,J=26,SAB,A=2,B=2,PLUS,CRY,D=2>        ; 3101
000505' 102021  000700
000506' 310250  000000                  MWORD   <JMAP,J=5000>                               ; 3102
000507' 000000  000040

                                ; Data pattern floating 1's - 100,,000000

000510' 320000  000000                  MWORD   <ADDR=3200,CONT,SDO,B=2,AND,D=2>            ; 3200
000511' 742001  000340
000512' 320100  270000                  MWORD   <LDCT,J=27,SAB,A=2,B=2,PLUS,CRY,D=2>        ; 3201
000513' 102021  000700
000514' 320250  000000                  MWORD   <JMAP,J=5000>                               ; 3202
000515' 000000  000040

                                ; Data pattern floating 1's - 200,,000000

000516' 330000  000000                  MWORD   <ADDR=3300,CONT,SDO,B=2,AND,D=2>            ; 3300
000517' 742001  000340
000520' 330100  300000                  MWORD   <LDCT,J=30,SAB,A=2,B=2,PLUS,CRY,D=2>        ; 3301
000521' 102021  000700
000522' 330250  000000                  MWORD   <JMAP,J=5000>                               ; 3302
000523' 000000  000040

                                ; Data pattern floating 1's - 400,,000000

000524' 340000  000000                  MWORD   <ADDR=3400,CONT,SDO,B=2,AND,D=2>            ; 3400
000525' 742001  000340
000526' 340100  310000                  MWORD   <LDCT,J=31,SAB,A=2,B=2,PLUS,CRY,D=2>        ; 3401
000527' 102021  000700
000530' 340250  000000                  MWORD   <JMAP,J=5000>                               ; 3402
000531' 000000  000040

                                ; Data pattern floating 1's - 1000,,000000

000532' 350000  000000                  MWORD   <ADDR=3500,CONT,SDO,B=2,AND,D=2>            ; 3500
000533' 742001  000340
000534' 350100  320000                  MWORD   <LDCT,J=32,SAB,A=2,B=2,PLUS,CRY,D=2>        ; 3501
000535' 102021  000700
000536' 350250  000000                  MWORD   <JMAP,J=5000>                               ; 3502
000537' 000000  000040

```

```
                                   ; Data pattern floating 1's - 2000,,000000

000540' 360000  000000                     MWORD   <ADDR=3600,CONT,SDO,B=2,AND,D=2>          ; 3600
000541' 742001  000340
000542' 360100  330000                     MWORD   <LDCT,J=33,SAB,A=2,B=2,PLUS,CRY,D=2>      ; 3601
000543' 102021  000700
000544' 360250  000000                     MWORD   <JMAP,J=5000>                             ; 3602
000545' 000000  000040

                                   ; Data pattern floating 1's - 4000,,000000

000546' 370000  000000                     MWORD   <ADDR=3700,CONT,SDO,B=2,AND,D=2>          ; 3700
000547' 742001  000340
000550' 370100  340000                     MWORD   <LDCT,J=34,SAB,A=2,B=2,PLUS,CRY,D=2>      ; 3701
000551' 102021  000700
000552' 370250  000000                     MWORD   <JMAP,J=5000>                             ; 3702
000553' 000000  000040

                                   ; Data pattern floating 1's - 10000,,000000

000554' 400000  000000                     MWORD   <ADDR=4000,CONT,SDO,B=2,AND,D=2>          ; 4000
000555' 742001  000340
000556' 400100  350000                     MWORD   <LDCT,J=35,SAB,A=2,B=2,PLUS,CRY,D=2>      ; 4001
000557' 102021  000700
000560' 400250  000000                     MWORD   <JMAP,J=5000>                             ; 4002
000561' 000000  000040

                                   ; Data pattern floating 1's - 20000,,000000

000562' 410000  000000                     MWORD   <ADDR=4100,CONT,SDO,B=2,AND,D=2>          ; 4100
000563' 742001  000340
000564' 410100  360000                     MWORD   <LDCT,J=36,SAB,A=2,B=2,PLUS,CRY,D=2>      ; 4101
000565' 102021  000700
000566' 410250  000000                     MWORD   <JMAP,J=5000>                             ; 4102
000567' 000000  000040

                                   ; Data pattern floating 1's - 40000,,000000

000570' 420000  000000                     MWORD   <ADDR=4200,CONT,SDO,B=2,AND,D=2>          ; 4200
000571' 742001  000340
000572' 420100  370000                     MWORD   <LDCT,J=37,SAB,A=2,B=2,PLUS,CRY,D=2>      ; 4201
000573' 102021  000700
000574' 420250  000000                     MWORD   <JMAP,J=5000>                             ; 4202
000575' 000000  000040

                                   ; Data pattern floating 1's - 100000,,000000

000576' 430000  000000                     MWORD   <ADDR=4300,CONT,SDO,B=2,AND,D=2>          ; 4300
000577' 742001  000340
000600' 430100  400000                     MWORD   <LDCT,J=40,SAB,A=2,B=2,PLUS,CRY,D=2>      ; 4301
000601' 102021  000700
000602' 430250  000000                     MWORD   <JMAP,J=5000>                             ; 4302
000603' 000000  000040

                                   ; Data pattern floating 1's - 200000,,000000
```

```
                                        MWORD   <ADDR=4400,CONT,SDO,B=2,AND,D=2>        ; 4400
000604' 440000  000000
000605' 742001  000340                  MWORD   <LDCT,J=41,SAB,A=2,B=2,PLUS,CRY,D=2>    ; 4401
000606' 440100  410000
000607' 102021  000700                  MWORD   <JMAP,J=5000>                           ; 4402
000610' 440250  000000
000611' 000000  000040

                                ; Data pattern floating 1's - 400000,,000000

                                        MWORD   <ADDR=4500,CONT,SDO,B=2,AND,D=2>        ; 4500
000612' 450000  000000
000613' 742001  000340                  MWORD   <LDCT,J=40,SAB,A=2,B=2,PLUS,CRY,D=2>    ; 4501
000614' 450100  400000
000615' 102021  000700                  MWORD   <RPCT,J=4502,SAB,A=2,B=2,PLUS,D=2>      ; 4502
000616' 450245  020000
000617' 102021  000220                  MWORD   <JMAP,J=5001,SAB,A=2,B=2,D=7>           ; 4503
000620' 450350  010000
000621' 107021  000040

                                ; Data pattern 0's

                                        MWORD   <ADDR=4600,JMAP,J=5001,SDO,B=2,AND,D=2> ; 4600
000622' 460050  010000
000623' 742001  000040

                                ; Subroutine to write data to formatter, read back, and place in EBUF

                                        MWORD   <ADDR=5000,RPCT,J=5000,SAB,A=2,B=2,PLUS,D=2> ; 5000
000624' 500050  000000
000625' 102021  000220                  MWORD   <CONT,SOA,A=2,D=1,OENA,SELF,MGC=40>     ; 5001
000626' 500100  002040
000627' 401020  003340                  MWORD   <CONT,SDO,B=2,D=2,SELM,MGC=200>         ; 5002
000630' 500200  000200
000631' 702001  002340                  MWORD   <CONT,SOA,A=2,D=1,OENA,SELE,MGC=4>      ; 5003
000632' 500300  002004
000633' 401020  005340                  MWORD   <JMAP,J=5004>                           ; 5004
000634' 500450  040000
000635' 000000  000040
000636' 777777  777777                  -1
```

```

                                   ;#*********************************************************************
                                   ;* Test 2 - PLI Buffer Data Loopback
                                   ;
                                   ; Description:  This test does simple data loopback over the MBUS
                                   ;               through the CBUF, through the PMUX,  through  the
                                   ;               PLI Output Buffer, through the PLI Input  Buffer,
                                   ;               through the CMUX, and back to the MBUS.
                                   ;
                                   ; Procedure:    Port Clear
                                   ;               Set RAR to 0
                                   ;               Execute single steps
                                   ;               Read EBUF, verify data is correct
                                   ;
                                   ;               Repeat with each data pattern:
                                   ;                 0's
                                   ;                 1's
                                   ;                 floating 1's
                                   ;                 floating 0's
                                   ;
                                   ; Failure:      ---
                                   ;#*********************************************************************

                                   ; Test data

 000637' 254 00 0 00 000646'       TSTC2:  JRST    TG2                     ; go start test
 000640' 100604  000002                    CBUS!NDMP!NDCB!ZCBUS!2          ; test mask
 000641' 000717' 012733'                   T2M,,[ASCIZ ^PLI Buffer Data Loopback^]
 000642' 012740' 012746'                   [EXP C13,C22,C21,C20,C23,MLAST!C24],,[EXP C12,C19,C8,C3,C4,MLAST!C2]
 000643' 000000  003034'                   TSTC11                          ; failure test table
 000644' 000000000000#                     TSTC40                          ;   ...
 000645' 777777  777777                    -1

                                   ; Start test

 000646' 201 00 0 00 000000'       TG2:    MOVEI   Z9                      ; get address of module start
 000647' 260 17 0 00 000012*               GO      TRACE                   ; handle trace output
 000650' 201 01 0 00 000717'               MOVEI   1,T2M                   ; set up microcode address
 000651' 260 17 0 00 000014*               GO      TLOAD                   ; load/verify it
 000652' 263 17 0 00 000000                RTN                             ; failed - exit test

                                   ; Initialization

 000653' 400 15 0 00 000000        TL2:    SETZ    ERFLG,                  ; clear error flag
 000654' 260 17 0 00 000017*               GO      IPACLR                  ; clear port
 000655' 402 00 0 00 000020*               SETZM   TSTSUB                  ; initialize subtest number
 000656' 201 06 0 00 000670'               MOVEI   6,TS2                   ; get sstep table address
```

```

                                       ; Loop on single step table entries

000657' 260 17 0 00 000022*            TA2:    GO      BEXEC                   ; execute table entry
000660' 254 00 0 00 000667'                    JRST    TX2                     ; end of sstep table
000661' 254 00 0 00 000657'                    JRST    TA2                     ; keep looping after call
000662' 474 15 0 00 000000                     SETO    ERFLG,                  ; error - EBUF data incorrect

                                       ; Handle error printouts and scope looping

000663' 027 00 0 00 000715'                    SCOPER  MA2                     ; print error message
000664' 254 00 0 00 000653'                    JRST    TL2                     ; loop on error
000665' 254 00 0 00 000667'                    JRST    TX2                     ; altmode exit
000666' 322 15 0 00 000657'                    JUMPE   ERFLG,TA2               ; do next sstep table entry

                                       ; End of test

000667' 263 17 0 00 000000             TX2:    RTN                             ; return

                                       ; Sstep table, with EBUF data, as:  (CMD,#Steps,Start addr,End Addr,EBUF data)

000670' 100700  005005                 TS2:    ATABLE  (SSSTRT,7,0,5005,377)
000671' 000000  000377
000672' 100700  015005                         ATABLE  (SSSTRT,7,1,5005,1)
000673' 000000  000001
000674' 101100  025005                         ATABLE  (SSSTRT,^D9,2,5005,2)
000675' 000000  000002
000676' 101200  035005                         ATABLE  (SSSTRT,^D10,3,5005,4)
000677' 000000  000004
000700' 101300  045005                         ATABLE  (SSSTRT,^D11,4,5005,10)
000701' 000000  000010
000702' 101400  055005                         ATABLE  (SSSTRT,^D12,5,5005,20)
000703' 000000  000020
000704' 101500  065005                         ATABLE  (SSSTRT,^D13,6,5005,40)
000705' 000000  000040
000706' 101600  075005                         ATABLE  (SSSTRT,^D14,7,5005,100)
000707' 000000  000100
000710' 101700  105005                         ATABLE  (SSSTRT,^D15,10,5005,200)
000711' 000000  000200
000712' 100600  115005                         ATABLE  (SSSTRT,6,11,5005,0)
000713' 000000  000000
000714' 000000  000000                         ATABLE  (SSLAST)

                                       ; Error messages

000715' 140000  012754'                MA2:    MSG!TXNOT![ASCIZ /Data loopback through PLI Buffers failed (result in EBUF)/]
000716' 000000000000#                          LAST!CALL!TXALL!BBPNT           ; print sequencer data
```

```
                                    ; Microcode

000717' 000001  000000              T2M:    MWORD   <ADDR=0,JMAP,J=100>             ; 0 - 1's
000720' 000000  000040
000721' 000102  000000                      MWORD   <JMAP,J=200>                    ; 1 - 1
000722' 000000  000040
000723' 000203  000000                      MWORD   <JMAP,J=300>                    ; 2 - 2
000724' 000000  000040
000725' 000304  000000                      MWORD   <JMAP,J=400>                    ; 3 - 4
000726' 000000  000040
000727' 000405  000000                      MWORD   <JMAP,J=500>                    ; 4 - 10
000730' 000000  000040
000731' 000506  000000                      MWORD   <JMAP,J=600>                    ; 5 - 20
000732' 000000  000040
000733' 000607  000000                      MWORD   <JMAP,J=700>                    ; 6 - 40
000734' 000000  000040
000735' 000710  000000                      MWORD   <JMAP,J=1000>                   ; 7 - 100
000736' 000000  000040
000737' 001011  000000                      MWORD   <JMAP,J=1100>                   ; 10 - 200
000740' 000000  000040
000741' 001112  000000                      MWORD   <JMAP,J=1200>                   ; 11 - 0's
000742' 000000  000040


                                    ; Data pattern 1's

000743' 010000  000000                      MWORD   <ADDR=100,CONT,SDO,B=2,AND,D=2>         ; 100
000744' 742001  000340
000745' 010150  010000                      MWORD   <JMAP,J=5001,SAB,A=2,B=2,XNOR,D=2>      ; 101
000746' 172021  000040

                                    ; Data pattern floating 1's - 1

000747' 020000  000000                      MWORD   <ADDR=200,CONT,SDO,B=2,AND,D=2>         ; 200
000750' 742001  000340
000751' 020150  010000                      MWORD   <JMAP,J=5001,SAB,A=2,B=2,PLUS,CRY,D=2>  ; 201
000752' 102021  000440

                                    ; Data pattern floating 1's - 2

000753' 030000  000000                      MWORD   <ADDR=300,CONT,SDO,B=2,AND,D=2>         ; 300
000754' 742001  000340
000755' 030100  000000                      MWORD   <LDCT,J=0,SAB,A=2,B=2,PLUS,CRY,D=2>     ; 301
000756' 102021  000700
000757' 030250  000000                      MWORD   <JMAP,J=5000>                           ; 302
000760' 000000  000040

                                    ; Data pattern floating 1's - 4

000761' 040000  000000                      MWORD   <ADDR=400,CONT,SDO,B=2,AND,D=2>         ; 400
000762' 742001  000340
000763' 040100  010000                      MWORD   <LDCT,J=1,SAB,A=2,B=2,PLUS,CRY,D=2>     ; 401
000764' 102021  000700
000765' 040250  000000                      MWORD   <JMAP,J=5000>                           ; 402
000766' 000000  000040
```

```


                                    ; Data pattern floating 1's - 10

000767' 050000  000000                      MWORD   <ADDR=500,CONT,SD0,B=2,AND,D=2>             ; 500
000770' 742001  000340
000771' 050100  020000                      MWORD   <LDCT,J=2,SAB,A=2,B=2,PLUS,CRY,D=2>         ; 501
000772' 102021  000700
000773' 050250  000000                      MWORD   <JMAP,J=5000>                               ; 502
000774' 000000  000040


                                    ; Data pattern floating 1's - 20

000775' 060000  000000                      MWORD   <ADDR=600,CONT,SD0,B=2,AND,D=2>             ; 600
000776' 742001  000340
000777' 060100  030000                      MWORD   <LDCT,J=3,SAB,A=2,B=2,PLUS,CRY,D=2>         ; 601
001000' 102021  000700
001001' 060250  000000                      MWORD   <JMAP,J=5000>                               ; 602
001002' 000000  000040


                                    ; Data pattern floating 1's - 40

001003' 070000  000000                      MWORD   <ADDR=700,CONT,SD0,B=2,AND,D=2>             ; 700
001004' 742001  000340
001005' 070100  040000                      MWORD   <LDCT,J=4,SAB,A=2,B=2,PLUS,CRY,D=2>         ; 701
001006' 102021  000700
001007' 070250  000000                      MWORD   <JMAP,J=5000>                               ; 702
001010' 000000  000040


                                    ; Data pattern floating 1's - 100

001011' 100000  000000                      MWORD   <ADDR=1000,CONT,SD0,B=2,AND,D=2>            ; 1000
001012' 742001  000340
001013' 100100  050000                      MWORD   <LDCT,J=5,SAB,A=2,B=2,PLUS,CRY,D=2>         ; 1001
001014' 102021  000700
001015' 100250  000000                      MWORD   <JMAP,J=5000>                               ; 1002
001016' 000000  000040


                                    ; Data pattern floating 1's - 200

001017' 110000  000000                      MWORD   <ADDR=1100,CONT,SD0,B=2,AND,D=2>            ; 1100
001020' 742001  000340
001021' 110100  060000                      MWORD   <LDCT,J=6,SAB,A=2,B=2,PLUS,CRY,D=2>         ; 1101
001022' 102021  000700
001023' 110250  000000                      MWORD   <JMAP,J=5000>                               ; 1102
001024' 000000  000040


                                    ; Data pattern 0's

001025' 120050  010000                      MWORD   <ADDR=1200,JMAP,J=5001,SD0,B=2,AND,D=2> ; 1200
001026' 742001  000040


                                    ; Subroutine to write data to PLI Buffers, read back, and place in EBUF

001027' 500050  000000                      MWORD   <ADDR=5000,RPCT,J=5000,SAB,A=2,B=2,PLUS,D=2> ; 5000
001030' 102021  000220
```

```
   824  001031' 500100  002040          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELM,MGC=40>  ; 5001
   825  001032' 431020  002340
   826  001033' 500200  000006          MWORD   <CONT,SDO,D=1,SELP,MGC=6>               ; 5002
   827  001034' 701000  001340
   828  001035' 500300  000300          MWORD   <CONT,SDO,OR,B=2,D=2,SELM,MGC=300>      ; 5003
   829  001036' 732001  002340
   830  001037' 500400  002004          MWORD   <CONT,SOA,A=2,D=1,OENA,SELE,MGC=4>      ; 5004
   831  001040' 401020  005340
   832  001041' 500550  050000          MWORD   <JMAP,J=5005>                           ; 5005
   833  001042' 000000  000040
   834  001043' 777777  777777          -1
```

```

                                   ;#*****************************************************************
                                   ;* Test 3 - CBUS to EBUF Data Transfer
                                   ;
                                   ; Description:  This test performs a simple single word CBUS data
                                   ;               transfer from KL10 memory over the  CBUS  through
                                   ;               the CBUS module over the MBUS through  the 2901's
                                   ;               to the EBUF.  The EBUF is then read and verified.
                                   ;
                                   ; Procedure:    KL> Port Clear
                                   ;               KL> Set up a CCW list to transfer 1 word
                                   ;               KL> Start up port
                                   ;               UC> Read word from CBUS and put in EBUF
                                   ;               KL> Stop microcode, read EBUF
                                   ;               KL> Verify data transfer completed ok
                                   ;
                                   ;               Repeat with data patterns 1-114
                                   ;
                                   ; Failure:      ---
                                   ;#*****************************************************************

                                   ; Test data

001044' 254 00 0 00 001055'        TSTC3:  JRST    TG3                             ; go start test
001045' 100604  000003                     CBUS!NDMP!NDCB!ZCBUS!3                  ; test mask
001046' 001123' 012770'                    T3M,,[ASCIZ ^CBUS to EBUF Data Transfer^]
001047' 012776' 013000'                    [EXP C1,MLAST!C14],,[EXP C5,C4,C8,MLAST!C2]
001050' 000000  001152'                    TSTC4                                   ; failure test table
001051' 000000  001275'                    TSTC5                                   ;   ...
001052' 000000000000#                      TSTC40
001053' 000000000000#                      TSTC41
001054' 777777  777777                     -1

                                   ; Start test

001055' 201 00 0 00 000000'        TG3:    MOVEI   Z9                              ; get address of module start
001056' 260 17 0 00 000647*                GO      TRACE                           ; handle trace output
001057' 201 01 0 00 001123'                MOVEI   1,T3M                           ; set up microcode address
001060' 260 17 0 00 000651*                GO      TLOAD                           ; load/verify it
001061' 263 17 0 00 000000                 RTN                                     ; failed - exit test

                                   ; Initialization

001062' 400 15 0 00 000000         TL3:    SETZ    ERFLG,                          ; clear error flag
001063' 201 06 0 00 001100'                MOVEI   6,TS3                           ; get test table address
001064' 402 00 0 00 000655*                SETZM   TSTSUB                          ; initialize subtest number
001065' 476 00 0 00 000000*                SETOM   CSTATF                          ; set check logout status flag
```

```

                                        ; Loop on test execute table entries

001066' 260 17 0 00 000654*             TA3:    GO      IPACLR                  ; clear port
001067' 260 17 0 00 000000*             TB3:    GO      CEXEC                   ; execute table entry
001070' 254 00 0 00 001077'                     JRST    TX3                     ; end of test execute table
001071' 254 00 0 00 001067'                     JRST    TB3                     ; keep looping after call
001072' 474 15 0 00 000000                      SETO    ERFLG,                  ; error occurred

                                        ; Handle error printouts and scope looping

001073' 027 00 0 00 001121'                     SCOPER  MA3                     ; print error message
001074' 254 00 0 00 001062'                     JRST    TL3                     ; loop on error
001075' 254 00 0 00 001077'                     JRST    TX3                     ; altmode exit
001076' 322 15 0 00 001066'                     JUMPE   ERFLG,TA3               ; do next test execute table entry

                                        ; End of test

001077' 263 17 0 00 000000              TX3:    RTN                             ; return

                                        ; Data Test Execute Table, as:  (CMD,parameters)

001100' 240000  001110'                 TS3:    CTABLE  (CCALL,TS3SET)          ; set up initial data pattern
001101' 300000  001112'                 TS3A:   CTABLE  (CCALLC,TS3INC)         ; increment data pattern
001102' 100000  000001                          CTABLE  (CSETWR,0,1)            ; set up write (default pattern,1 word)
001103' 240000  001116'                         CTABLE  (CCALL,TS3SEE)          ; set up correct EBUF contents
001104' 200000  000000                          CTABLE  (CSTART,0)              ; start up port
001105' 340001  014064'                         CTABLE  (CWAIT,1,TS3EBU)        ; check if completed (and verify EBUF)
001106' 440000  001101'                         CTABLE  (CJRST,TS3A)            ; loop till done
001107' 000000  000000                          CTABLE  (CLAST)                 ; end of table

                                        ; Initialize data pattern selection

001110' 402 00 0 00 000000*             TS3SET: SETZM   CPAT                    ; clear data pattern
001111' 263 17 0 00 000000                      RTN                             ; return

                                        ; Increment data pattern

001112' 350 01 0 00 001110*             TS3INC: AOS     1,CPAT                  ; increment data pattern
001113' 307 01 0 00 000114                      CAIG    1,114                   ; done with test yet?
001114' 350 00 0 17 000000                      AOS     (P)                     ; no - set up proper return
001115' 263 17 0 00 000000                      RTN                             ; return

                                        ; Set up correct EBUF data

001116' 200 00 0 00 000000*             TS3SEE: MOVE    BUFF                    ; get correct EBUF data
001117' 202 00 0 00 014064'                     MOVEM   TS3EBU#                 ; save it
001120' 263 17 0 00 000000                      RTN                             ; return

                                        ; Error messages

001121' 140000  013004'                 MA3:    MSG!TXNOT![ASCIZ /Data transfer test error detected (result in EBUF)/]
001122' 000000000000#                           LAST!CALL!TXALL!CCPNT           ; print test data
```

```

                                   ; Microcode

001123' 000000  010000             T3M:   MWORD   <ADDR=0,JMAP,J=1,SDO,AND,D=2>           ; 0
001124' 742000  000040

                                   ;      First, start the CBUS

001125' 000100  000200                    MWORD   <CONT,D=1,SELC,MGC=200>                 ; 1
001126' 001000  004340

                                   ;      Then wait for CBUS to be available (1 word done)

001127' 000200  040000                    MWORD   <CJP,J=4,D=1,CENA,CCCA>                 ; 2
001130' 001400  000060
001131' 000300  020000                    MWORD   <JMAP,J=2,D=1>                          ; 3
001132' 001000  000040

                                   ;      Now move it to formatter then to 2901's then to EBUF

001133' 000400  000004                    MWORD   <CONT,D=1,SELC,MGC=4>                   ; 4
001134' 001000  004340
001135' 000500  000200                    MWORD   <CONT,SDO,OR,D=2,SELM,MGC=200>          ; 5
001136' 732000  002340
001137' 000600  002004                    MWORD   <CONT,SOA,OR,D=1,OENA,SELE,MGC=4>       ; 6
001140' 431000  005340

                                   ;      Finally set STOP and STORE

001141' 000700  000140                    MWORD   <CONT,SELC,MGC=140>                     ; 7
001142' 001000  004340
001143' 001000  100000                    MWORD   <JMAP,J=10>                             ; 10
001144' 000000  000040

                                   ;      Initialization step

001145' 500050  010100                    MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>    ; 5000
001146' 000000  004040
001147' 500150  010020                    MWORD   <JMAP,J=5001,SELM,MGC=20>               ; 5001
001150' 000000  002040
001151' 777777  777777                    -1
```

```

                                             ;#*********************************************************************
                                             ;* Test 4 - 2901 to CBUS Data Transfer
                                             ;
                                             ; Description:  This test performs a simple single word CBUS data
                                             ;               transfer from the 2901's across the MBUS  to  the
                                             ;               CBUS module, then to KL10 memory. The data buffer
                                             ;               contents is then verified.
                                             ;
                                             ; Procedure:    KL> Port Clear
                                             ;               KL> Set up a CCW list to transfer 1 word
                                             ;               KL> Start up port
                                             ;               UC> Generate a data pattern in 2901 and
                                             ;                   transfer over the CBUS to 10 memory
                                             ;               KL> Verify the data is correct
                                             ;               KL> Verify data transfer completed ok
                                             ;
                                             ;               Repeat with data patterns 0's and 1's
                                             ;
                                             ; Failure:      ---
                                             ;#*********************************************************************

                                             ; Test data

001152' 254 00 0 00 001162'                  TSTC4:  JRST    TG4                     ; go start test
001153' 100604  000004                               CBUS!NDMP!NDCB!ZCBUS!4          ; test mask
001154' 001220' 013017'                              T4M,,[ASCIZ ^2901 to CBUS Data Transfer^]
001155' 013025' 013000'                              [EXP C1,MLAST!C15],,[EXP C5,C4,C8,MLAST!C2]
001156' 000000  001275'                              TSTC5                           ; failure test table
001157' 000000000000#                                TSTC40                          ;   ...
001160' 000000000000#                                TSTC41
001161' 777777  777777                               -1

                                             ; Start test

001162' 201 00 0 00 000000'                  TG4:    MOVEI   Z9                      ; get address of module start
001163' 260 17 0 00 001056*                          GO      TRACE                   ; handle trace output
001164' 201 01 0 00 001220'                          MOVEI   1,T4M                   ; set up microcode address
001165' 260 17 0 00 001060*                          GO      TLOAD                   ; load/verify it
001166' 263 17 0 00 000000                           RTN                             ; failed - exit test

                                             ; Initialization

001167' 400 15 0 00 000000                   TL4:    SETZ    ERFLG,                  ; clear error flag
001170' 201 06 0 00 001206'                          MOVEI   6,TS4                   ; get test table address
001171' 402 00 0 00 001064*                          SETZM   TSTSUB                  ; initialize subtest number
001172' 476 00 0 00 001065*                          SETOM   CSTATF                  ; set check logout status flag
001173' 400 15 0 00 000000                           SETZ    ERFLG,                  ; clear error flag
```

```

                                         ; Loop on test execute table entries

001174' 260 17 0 00 001066*              TA4:    GO      IPACLR                  ; clear port
001175' 260 17 0 00 001067*              TB4:    GO      CEXEC                   ; execute table entry
001176' 254 00 0 00 001205'                        JRST    TX4                   ; end of test execute table
001177' 254 00 0 00 001175'                        JRST    TB4                   ; keep looping after call
001200' 474 15 0 00 000000                         SETO    ERFLG,                ; error occurred

                                         ; Handle error printouts and scope looping

001201' 027 00 0 00 001216'                      SCOPER  MA4                     ; print error message
001202' 254 00 0 00 001167'                      JRST    TL4                     ; loop on error
001203' 254 00 0 00 001205'                      JRST    TX4                     ; altmode exit
001204' 322 15 0 00 001174'                      JUMPE   ERFLG,TA4               ; do next test execute table entry

                                         ; End of test

001205' 263 17 0 00 000000               TX4:    RTN                             ; return

                                         ; Data Test Execute Table, as:  (CMD,parameters)

001206' 040000  000001                   TS4:    CTABLE  (CSETRD,0,1)            ; set up write (default pattern,1 word)
001207' 200000  000000                           CTABLE  (CSTART,0)              ; start up port at location 0
001210' 340001  000000                           CTABLE  (CWAIT,1)               ; check if completed
001211' 500000  000000                           CTABLE  (CEXIT)                 ; exit if error yet
001212' 040000  000001                           CTABLE  (CSETRD,0,1)            ; set up write (default pattern,1 word)
001213' 200000  000001                           CTABLE  (CSTART,1)              ; start up port at location 1
001214' 340001  000000                           CTABLE  (CWAIT,1)               ; check if completed
001215' 000000  000000                           CTABLE  (CLAST)                 ; end of table

                                         ; Error messages

001216' 140000  013027'                  MA4:    MSG!TXNOT![ASCIZ /Data transfer test error detected/]
001217' 000000000000#                            LAST!CALL!TXALL!CCPNT           ; print test data

                                         ; Microcode

001220' 000000  030000                   T4M:    MWORD   <ADDR=0,JMAP,J=3,SDO,AND,D=2>          ; 0
001221' 742000  000040
001222' 000100  020000                           MWORD   <JMAP,J=2,SDO,AND,D=2>                 ; 1
001223' 742000  000040
001224' 000200  030000                           MWORD   <JMAP,J=3,SAB,XNOR,D=2>                ; 2
001225' 172000  000040

                                         ;       First, load the formatter

001226' 000300  002040                           MWORD   <CONT,SOA,D=1,OENA,SELF,MGC=40>        ; 3
001227' 401000  003340

                                         ;       Then, start the CBUS

001230' 000400  000220                           MWORD   <CONT,D=1,SELC,MGC=220>                ; 4
001231' 001000  004340

```

```
                                ;       Then, wait for CBUS to be available

001232' 000500  070000                  MWORD   <CJP,J=7,D=1,CENA,CCCA>                 ; 5
001233' 001400  000060
001234' 000600  050000                  MWORD   <JMAP,J=5,D=1>                          ; 6
001235' 001000  000040

                                ;       Put data in CB Output Buffer

001236' 000700  000002                  MWORD   <CONT,D=1,SELC,MGC=2>                   ; 7
001237' 001000  004340

                                ;       Then, wait for CBUS to be available (1 word done)

001240' 001000  120000                  MWORD   <CJP,J=12,D=1,CENA,CCCA>                ; 10
001241' 001400  000060
001242' 001100  100000                  MWORD   <JMAP,J=10,D=1>                         ; 11
001243' 001000  000040

                                ;       Finally set STOP and STORE

001244' 001200  000140                  MWORD   <CONT,SELC,MGC=140>                     ; 12
001245' 000000  004340
001246' 001300  130000                  MWORD   <JMAP,J=13>                             ; 13
001247' 000000  000040

                                ;       Finally, write bit 'RspQAV' in CSR to indicate we are done,
                                ;       and write word count to EBUF

001250' 001400  310200                  MWORD   <LDCT,J=31,SDO,OR,B=17,D=2,SKCN,MGC=200>        ; 14
001251' 732007  640300
001252' 001500  150000                  MWORD   <RPCT,J=15,SOB,OR,B=17,D=5>                     ; 15
001253' 335007  400220
001254' 001600  002004                  MWORD   <CONT,SOA,A=17,OR,D=1,OENA,SELE,MGC=4>          ; 16
001255' 431170  005340
001256' 001700  210040                  MWORD   <CJP,J=21,CENA,CCGC,SELE,MGC=40>                ; 17
001257' 000400  015060
001260' 002000  170000                  MWORD   <JMAP,J=17>                                     ; 20
001261' 000000  000040
001262' 002100  000200                  MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>              ; 21
001263' 431010  005340
001264' 002200  002004                  MWORD   <CONT,SOA,OR,A=5,D=1,OENA,SELE,MGC=4>           ; 22
001265' 431050  005340
001266' 002300  230000                  MWORD   <JMAP,J=23>                                     ; 23
001267' 000000  000040

                                ;       Initialization step

001270' 500050  010100                  MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>    ; 5000
001271' 000000  004040
001272' 500150  010020                  MWORD   <JMAP,J=5001,SELM,MGC=20>               ; 5001
001273' 000000  002040
001274' 777777  777777                  -1
```

```

                                               ;#*******************************************************************
                                               ;* Test 5 - CBUS to 2901 Multiple word Transfer
                                               ;
                                               ; Description:  This test performs a lengthy CBUS  data  transfer
                                               ;               from the KL10 memory across CBUS to the port. The
                                               ;               microcode accepts the first data word as the data
                                               ;               pattern to be used, and  verifies  the  remaining
                                               ;               words against this.
                                               ;
                                               ; Procedure:    KL> Port Clear
                                               ;               KL> Set up a CCW list to transfer 102400 words
                                               ;                   (512 words in user mode)
                                               ;               KL> Start up port
                                               ;               UC> Initialize word count
                                               ;               UC> Read first word from CBUS and put in 2901 R4
                                               ;               UC> Repeat until 'Last Wd' is seen:
                                               ;                     Read a word
                                               ;                     Compare to first word
                                               ;                       If error - put word in error into R3
                                               ;                                - put word count in EBUF
                                               ;                                - abort test
                                               ;                       If ok - do next word
                                               ;               UC> Assert STOP and STORE and put word count into EBUF
                                               ;               KL> Stop microcode, read EBUF - verify word count
                                               ;               KL> Read R4 to get correct data (1st word read)
                                               ;               KL> If stopped at wrong location, read R3 to get
                                               ;                   actual data.
                                               ;               KL> Verify data transfer completed ok
                                               ;
                                               ;               Repeat with data patterns 1-114
                                               ;
                                               ; Failure:      ---
                                               ;#*******************************************************************

                                               ; Test data

        001275' 254 00 0 00 001305'           TSTC5:  JRST    TG5                             ; go start test
        001276' 100604  000005                        CBUS!NDMP!NDCB!ZCBUS!5                  ; test mask
        001277' 001453' 013036'                       T5M,,[ASCIZ ^CBUS to 2901 Multiple Word Transfer^]
        001300' 013046' 013051'                       [EXP C1,C14,MLAST!C17],,[EXP C12,C5,C8,C4,MLAST!C2]
        001301' 000000000000#                         TSTC40                                  ; failure test table
        001302' 000000000000#                         TSTC41                                  ;   ...
        001303' 000000  001612'                       TSTC6
        001304' 777777  777777                        -1

                                               ; Start test

        001305' 201 00 0 00 000000'           TG5:    MOVEI   Z9                              ; get address of module start
        001306' 260 17 0 00 001163*                   GO      TRACE                           ; handle trace output
        001307' 201 01 0 00 001453'                   MOVEI   1,T5M                           ; set up microcode address
        001310' 260 17 0 00 001165*                   GO      TLOAD                           ; load/verify it
        001311' 263 17 0 00 000000                    RTN                                     ; failed - exit test
```

```


                                    ; Initialization

 001312' 400 15 0 00 000000         TL5:    SETZ    ERFLG,                  ; clear error flag
 001313' 201 06 0 00 001330'                MOVEI   6,TS5                   ; get test table address
 001314' 402 00 0 00 001171*                SETZM   TSTSUB                  ; initialize subtest number
 001315' 400 15 0 00 000000                 SETZ    ERFLG,                  ; clear error flag

                                    ; Loop on test execute table entries

 001316' 260 17 0 00 001174*        TA5:    GO      IPACLR                  ; clear port
 001317' 260 17 0 00 001175*        TB5:    GO      CEXEC                   ; execute table entry
 001320' 254 00 0 00 001327'                 JRST   TX5                     ; end of test execute table
 001321' 254 00 0 00 001317'                 JRST   TB5                     ; keep looping after call
 001322' 474 15 0 00 000000                  SETO   ERFLG,                  ; error occurred

                                    ; Handle error printouts and scope looping

 001323' 027 00 0 00 001370'                SCOPER  MA5                     ; print error message
 001324' 254 00 0 00 001312'                JRST    TL5                     ; loop on error
 001325' 254 00 0 00 001327'                JRST    TX5                     ; altmode exit
 001326' 322 15 0 00 001316'                JUMPE   ERFLG,TA5               ; do next test execute table entry

                                    ; End of test

 001327' 263 17 0 00 000000         TX5:    RTN                             ; return

                                    ; Data Test Execute Table, as:  (CMD,parameters)

 001330' 240000  001340'            TS5:    CTABLE  (CCALL,TS5SET)          ; set up initial data pattern
 001331' 300000  001346'            TS5A:   CTABLE  (CCALLC,TS5INC)         ; increment data pattern
 001332' 100000  000000                     CTABLE  (CSETWR,0,0)            ; set up write
 001333' 200000  001000                     CTABLE  (CSTART,1000)           ; start up port
 001334' 341000  000000                     CTABLE  (CWAIT,1000,0)          ; check if completed
 001335' 240000  001356'                    CTABLE  (CCALL,TS5CHE)          ; check results
 001336' 440000  001331'                    CTABLE  (CJRST,TS5A)            ; loop till done
 001337' 000000  000000                     CTABLE  (CLAST)                 ; end of table

                                    ; Initialize data pattern selection

 001340' 402 00 0 00 001112*        TS5SET: SETZM   CPAT                    ; clear data pattern
 001341' 201 00 0 00 001000                 MOVEI   ^D512                   ; length of data transfer is
 001342' 336 00 0 00 030037                 SKIPN   USER                    ;     102400 words exec mode
 001343' 201 00 0 00 310000                 MOVEI   ^D102400                ;    and 512 words user mode
 001344' 202 00 0 00 000000*                MOVEM   CLEN                    ; save it
 001345' 263 17 0 00 000000                 RTN                             ; return
```

```

                                 ; Increment data pattern

001346' 350 01 0 00 001340*      TS5INC: AOS     1,CPAT                     ; increment data pattern
001347' 260 17 0 00 001316*              GO      IPACLR                     ; do a 'port clear'
001350' 307 01 0 00 000114               CAIG    1,114                      ; done with test yet?
001351' 350 00 0 17 000000               AOS     (P)                        ; no - set up proper return
001352' 332 00 0 00 000000*              SKIPE   UDEBUG                     ; debug mode?
                                         JRST    [CAIG    1,10              ; yes - limit to only 10 segments
                                                  AOS     (P)
001353' 254 00 0 00 013056'                       JRST    .+1]
001354' 476 00 0 00 001172*              SETOM   CSTATF                     ; set check logout status flag
001355' 263 17 0 00 000000               RTN                                ; return

                                 ; Check results

001356' 260 17 0 00 001373'      TS5CHE: GO      MA5CHE                     ; get additional data
001357' 200 01 0 00 000000*              MOVE    1,CEBUFA                   ; get word count
001360' 312 01 0 00 001344*              CAME    1,CLEN                     ; correct?
001361' 474 15 0 00 000000               SETO    ERFLG,                     ; no - set error flag
001362' 200 01 0 00 014011'              MOVE    1,SAVDA1                   ; get actual data
001363' 312 01 0 00 014012'              CAME    1,SAVDA2                   ; same as correct?
001364' 474 15 0 00 000000               SETO    ERFLG,                     ; no - set error flag
001365' 350 00 0 17 777772               AOS     -6(P)                      ; set up error return
001366' 350 00 0 17 777772               AOS     -6(P)                      ; set up error return
001367' 263 17 0 00 000000               RTN                                ; return

                                 ; Error messages

001370' 140000  013027'          MA5:    MSG!TXNOT![ASCIZ /Data transfer test error detected/]
001371' 240000  001373'                  CALL!TXNOT!MA5CHE                  ; check other results
001372' 000000000000#                    LAST!CALL!TXALL!CCPNT              ; print test data

                                 ; Check results - first the word count left in the EBUF

001373' 322 15 0 00 001406'      MA5CHE: JUMPE   ERFLG,MA5CHO               ; no error yet? - continue
001374' 200 01 0 00 001357*              MOVE    1,CEBUFA                   ; get word count
001375' 316 01 0 00 001360*              CAMN    1,CLEN                     ; correct?
001376' 254 00 0 00 001406'              JRST    MA5CHO                     ; yes - continue
001377' 331 00 0 00 000001               SKIPL   1                          ; word count error?
                                         JRST    [TMSGC  <Word count error detected by port:  >
                                                  TMSGC  <  Correct:>
                                                  PDEC   10,CLEN
                                                  TMSG   <.>
                                                  TMSGC  <  Actual:>
                                                  PDEC   11,CEBUFA              ; print it
                                                  TMSG   <.>
001400' 254 00 0 00 013100'                       JRST   MA5CHO]
001401' 037 00 0 00 013110'              TMSGC   <Parity error detected by port in word #>
001402' 210 00 0 00 001374*              MOVN    CEBUFA
001403' 350 00 0 00 000000               AOS
001404' 037 15 0 00 000000               PNTDEC
001405' 037 00 0 00 013074'              TMSG    <.>
```

```
                                        ; Then the correct/actual data being read

001406' 400 01 0 00 000000      MA5CHO: SETZ    1,                      ; ensure the port is
001407' 260 17 0 00 000000*             GO      LDCSR                   ;   stopped
001410' 201 01 0 00 000200              MOVEI   1,200                   ; get address to get R3
001411' 242 01 0 00 000001              LSH     1,1                     ; position correctly
001412' 260 17 0 00 000000*             GO      LDRAR                   ; load RAR register
001413' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; set up initial CSR data
001414' 260 17 0 00 001407*             GO      LDCSR                   ; write CSR register
001415' 201 01 0 00 200000              MOVEI   1,TSTEBF                ; set up CSR to read EBUF
001416' 260 17 0 00 001414*             GO      LDCSR                   ; write it
001417' 260 17 0 00 000000*             GO      RDEBUF                  ; get EBUF data (actual)
001420' 202 01 0 00 014011'             MOVEM   1,SAVDA1#               ; save it
001421' 400 01 0 00 000000              SETZ    1,                      ; ensure the port is
001422' 260 17 0 00 001416*             GO      LDCSR                   ;   stopped
001423' 201 01 0 00 000300              MOVEI   1,300                   ; get address to get R4
001424' 242 01 0 00 000001              LSH     1,1                     ; position correctly
001425' 260 17 0 00 001412*             GO      LDRAR                   ; load RAR register
001426' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; set up initial CSR data
001427' 260 17 0 00 001422*             GO      LDCSR                   ; write CSR register
001430' 201 01 0 00 200000              MOVEI   1,TSTEBF                ; set up CSR to read EBUF
001431' 260 17 0 00 001427*             GO      LDCSR                   ; write it
001432' 260 17 0 00 001417*             GO      RDEBUF                  ; get EBUF data (actual)
001433' 202 01 0 00 014012'             MOVEM   1,SAVDA2#               ; save it

                                ; Check for data compare error - and print if so

001434' 332 00 0 00 000015              SKIPE   ERFLG                   ; error yet?
001435' 316 01 0 00 014011'             CAMN    1,SAVDA1                ; yes - error?
001436' 263 17 0 00 000000              RTN                             ; no - return
001437' 037 00 0 00 013121'             TMSGC   <Data compare error - Word # >
001440' 200 00 0 00 001402*             MOVE    CEBUFA                  ; get word count
001441' 350 00 0 00 000000              AOS
001442' 037 15 0 00 000000              PNTDEC                          ; print it
001443' 037 00 0 00 013074'             TMSG    <.>
001444' 037 00 0 00 013130'             TMSGC   <  Correct: >
001445' 200 00 0 00 014012'             MOVE    SAVDA2                  ; get correct data
001446' 037 13 0 00 000000              PNTHW                           ; print it
001447' 037 00 0 00 013133'             TMSGC   <  Actual:  >
001450' 200 00 0 00 014011'             MOVE    SAVDA1                  ; get actual data
001451' 037 13 0 00 000000              PNTHW                           ; print it
001452' 263 17 0 00 000000              RTN                             ; return
```

```

                                ; Microcode - Read the first word

001453' 000001  000000          T5M:    MWORD   <ADDR=0,CJS,J=100,D=1>                          ; 0
001454' 001000  000020
001455' 000100  000000                  MWORD   <CONT,SOA,OR,A=3,B=4,D=2>                       ; 1
001456' 432032  000340
001457' 000200  000000                  MWORD   <CONT,SOA,AND,A=5,B=5,D=2>                      ; 2
001460' 442052  400340
001461' 000300  000000                  MWORD   <CONT,SOA,PLUS,A=5,B=5,D=2,CRY>                 ; 3
001462' 402052  400740

                                ;       Read 2nd through 102400 words

001463' 000401  030000                  MWORD   <CJS,J=103,D=1>                                 ; 4
001464' 001000  000020
001465' 000500  070000                  MWORD   <CJP,J=7,SAB,XOR,A=3,B=4,D=1,CENA,CCFZ>         ; 5
001466' 161432  020060
001467' 000600  110000                  MWORD   <JMAP,J=11>                                     ; 6
001470' 000000  000040
001471' 000700  000000                  MWORD   <CONT,SOB,B=5,PLUS,D=2,CRY>                     ; 7
001472' 302002  400740
001473' 001000  150000                  MWORD   <CJP,J=15,SOB,OR,B=12,D=2,CENA,CCMB>            ; 10
001474' 332405  120060
001475' 001100  040000                  MWORD   <JMAP,J=4>                                      ; 11
001476' 000000  000040

                                ;       Error (data compare) occurred

001477' 001200  000140                  MWORD   <CONT,SELC,MGC=140>                             ; 12
001500' 000000  004340
001501' 001300  002004                  MWORD   <CONT,SOA,OR,A=5,D=1,OENA,SELE,MGC=4>           ; 13
001502' 431050  005340
001503' 001400  140000                  MWORD   <JMAP,J=14>                                     ; 14
001504' 000000  000040

                                ;       All done -  set STOP and STORE

001505' 001500  000140                  MWORD   <CONT,SELC,MGC=140>                             ; 15
001506' 000000  004340

                                ;       Finally, write bit 'RspQAV' in CSR to indicate we are done,
                                ;       and write word count to EBUF

001507' 001600  310200                  MWORD   <LDCT,J=31,SDO,OR,B=17,D=2,SKCN,MGC=200>        ; 16
001510' 732007  640300
001511' 001700  170000                  MWORD   <RPCT,J=17,SOB,OR,B=17,D=5>                     ; 17
001512' 335007  400220
001513' 002000  002004                  MWORD   <CONT,SOA,A=17,OR,D=1,OENA,SELE,MGC=4>          ; 20
001514' 431170  005340
001515' 002100  230040                  MWORD   <CJP,J=23,CENA,CCGC,SELE,MGC=40>                ; 21
001516' 000400  015060
001517' 002200  210000                  MWORD   <JMAP,J=21>                                     ; 22
001520' 000000  000040
001521' 002300  000200                  MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>              ; 23
```

```
001522' 431010  005340                    MWORD   <CONT,SOA,OR,A=5,D=1,OENA,SELE,MGC=4>         ; 24
001523' 002400  002004
001524' 431050  005340                    MWORD   <JMAP,J=25>                                   ; 25
001525' 002500  250000
001526' 000000  000040

                                  ;       Routine to read a word from CBUS, through formatter into R3
                                  ;       (If this routine is started at 101, then the next word from
                                  ;       the CBUS is read).

001527' 010000  000200                    MWORD   <ADDR=100,CONT,SELC,MGC=200>                  ; 100
001530' 000000  004340
001531' 010100  100000                    MWORD   <LDCT,J=10>                                   ; 101
001532' 000000  000300
001533' 010201  020000                    MWORD   <RPCT,J=102>                                  ; 102
001534' 000000  000220
001535' 010301  050000                    MWORD   <CJP,J=105,CENA,CCCA>                         ; 103
001536' 000400  000060
001537' 010401  030000                    MWORD   <JMAP,J=103>                                  ; 104
001540' 000000  000040
001541' 010500  000000                    MWORD   <CONT>                                        ; 105
001542' 000000  000340
001543' 010600  000004                    MWORD   <CONT,SOB,AND,B=12,D=2,SELC,MGC=4>            ; 106
001544' 342005  004340
001545' 010700  000200                    MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>            ; 107
001546' 732001  402340
001547' 011001  120000                    MWORD   <CJP,J=112,SOB,XNOR,B=12,D=2,CENA,CCLW>       ; 110
001550' 372405  170060
001551' 011100  000000                    MWORD   <CONT,SOB,AND,B=12,D=2>                       ; 111
001552' 342005  000340
001553' 011201  150000                    MWORD   <CJP,J=115,CENA,CCCE>                         ; 112
001554' 000400  160060
001555' 011301  170000                    MWORD   <CJP,J=117,CENA,CCCP>                         ; 113
001556' 000400  140060
001557' 011400  000000                    MWORD   <CRTN>                                        ; 114
001560' 000000  000240

                                  ;       Error (channel error) occurred -  set STOP and STORE

001561' 011500  000140                    MWORD   <CONT,SELC,MGC=140>                           ; 115
001562' 000000  004340
001563' 011601  160000                    MWORD   <JMAP,J=116>                                  ; 116
001564' 000000  000040

                                  ;       Error (parity error) occurred

001565' 011700  000140                    MWORD   <CONT,SELC,MGC=140>                           ; 117
001566' 000000  004340
001567' 012000  002004                    MWORD   <CONT,SOA,XNOR,A=5,D=1,OENA,SELE,MGC=4>       ; 120
001570' 471050  005340
001571' 012101  210000                    MWORD   <JMAP,J=121>                                  ; 121
001572' 000000  000040

                                  ;       Routine to read R3 and place in EBUF

```

```
DFPTA9  MACRO %53B(1242) 09:42  1-Jul-85 Page 16-2
DFPTA9  MAC      18-Apr-85 11:00         CBUS MODULE TESTS

  1437  001573' 020002  012004                   MWORD   <ADDR=200,JMAP,J=201,SOA,A=3,OR,OENA,D=1,SELE,MGC=4>
  1438  001574' 431030  005040
  1439  001575' 020102  010000                   MWORD   <JMAP,J=201>                                  ; 201
  1440  001576' 000000  000040
  1441
  1442                                   ;       Routine to read R4 and place in EBUF
  1443
  1444  001577' 030003  012004                   MWORD   <ADDR=300,JMAP,J=301,SOA,A=4,OR,OENA,D=1,SELE,MGC=4>
  1445  001600' 431040  005040
  1446  001601' 030103  010000                   MWORD   <JMAP,J=301>                                  ; 201
  1447  001602' 000000  000040
  1448
  1449                                   ;       Initialization
  1450
  1451  001603' 100000  000020                   MWORD   <ADDR=1000,JZ,SELM,MGC=20>                    ; 1000
  1452  001604' 000000  002000
  1453
  1454                                   ;       Initialization step
  1455
  1456  001605' 500050  010100                   MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>          ; 5000
  1457  001606' 000000  004040
  1458  001607' 500150  010020                   MWORD   <JMAP,J=5001,SELM,MGC=20>                     ; 5001
  1459  001610' 000000  002040
  1460  001611' 777777  777777                   -1
```

```
DFPTA9  MACRO %53B(1242) 09:42  1-Jul-85 Page 17
DFPTA9  MAC      18-Apr-85 11:00         CBUS MODULE TESTS
```

```
                                   ;#*********************************************************************
                                   ;* Test 6 - 2901 to CBUS Multiple word Transfer
                                   ;
                                   ; Description:  This test performs a lengthy CBUS  data  transfer
                                   ;               from the port across the CBUS to KL10 memory. The
                                   ;               microcode builds each data pattern then transfers
                                   ;               this  data  to  the  KL10  until  the CCW list is
                                   ;               exhausted.  The data buffer is then verified.
                                   ;
                                   ; Procedure:    KL> Port Clear
                                   ;               KL> Load test microcode
                                   ;
                                   ;               KL> Load microcode to generate data pattern
                                   ;               KL> Set up a CCW list to transfer 102400 words
                                   ;                   (512 words in user mode)
                                   ;               KL> Start up port
                                   ;               UC> Initialize word count
                                   ;               UC> Write word to CBUS, and repeat this until done
                                   ;               UC> Assert STOP and STORE and put word count into EBUF
                                   ;               KL> Stop microcode, read EBUF - verify word count
                                   ;               KL> Read R4 to get correct data (1st word written)
                                   ;               KL> Verify data buffer
                                   ;
                                   ;               Repeat with data patterns 1-114
                                   ;#*********************************************************************

                                   ; Test data

001612' 254 00 0 00 001621'        TSTC6:  JRST    TG6                     ; go start test
001613' 100604  000006                     CBUS!NDMP!NDCB!ZCBUS!6          ; test mask
001614' 002022' 013136'                    T6M,,[ASCIZ ^2901 to CBUS Multiple Word Transfer^]
001615' 013146' 013051'                    [EXP C1,C15,MLAST!C16],,[EXP C12,C5,C8,C4,MLAST!C2]
001616' 000000000000#                      TSTC40                          ; failure test table
001617' 000000000000#                      TSTC41                          ;   ...
001620' 777777  777777                     -1

                                   ; Start test

001621' 201 00 0 00 000000'        TG6:    MOVEI   Z9                      ; get address of module start
001622' 260 17 0 00 001306*                GO      TRACE                   ; handle trace output
001623' 201 01 0 00 002022'                MOVEI   1,T6M                   ; set up microcode address
001624' 260 17 0 00 001310*                GO      TLOAD                   ; load/verify it
001625' 263 17 0 00 000000                 RTN                             ; failed - exit test

                                   ; Initialization

001626' 400 15 0 00 000000         TL6:    SETZ    ERFLG,                  ; clear error flag
001627' 201 06 0 00 001644'                MOVEI   6,TS6                   ; get test table address
001630' 402 00 0 00 001314*                SETZM   TSTSUB                  ; initialize subtest number
001631' 400 15 0 00 000000                 SETZ    ERFLG,                  ; clear error flag
```

```

                                        ; Loop on test execute table entries

   001632' 260 17 0 00 001347*          TA6:    GO      IPACLR                          ; clear port
   001633' 260 17 0 00 001317*          TB6:    GO      CEXEC                           ; execute table entry
   001634' 254 00 0 00 001643'                   JRST   TX6                             ; end of test execute table
   001635' 254 00 0 00 001633'                   JRST   TB6                             ; keep looping after call
   001636' 474 15 0 00 000000                    SETO   ERFLG,                          ; error occurred

                                        ; Handle error printouts and scope looping

   001637' 027 00 0 00 001742'                  SCOPER  MA6                             ; print error message
   001640' 254 00 0 00 001626'                  JRST    TL6                             ; loop on error
   001641' 254 00 0 00 001643'                  JRST    TX6                             ; altmode exit
   001642' 322 15 0 00 001632'                  JUMPE   ERFLG,TA6                       ; do next test execute table entry

                                        ; End of test

   001643' 263 17 0 00 000000           TX6:    RTN                                     ; return

                                        ; Data Test Execute Table, as:  (CMD,parameters)

   001644' 240000  001654'              TS6:    CTABLE  (CCALL,TS6SET)                  ; set up initial data pattern
   001645' 300000  001666'              TS6A:   CTABLE  (CCALLC,TS6INC)                 ; increment data pattern
   001646' 040000  000000                       CTABLE  (CSETRD,0,0)                    ; set up write
   001647' 200000  000100               TS6S:   CTABLE  (CSTART,100)                    ; start up port
   001650' 341000  000000                       CTABLE  (CWAIT,1000,0)                  ; check if completed
   001651' 240000  001730'                      CTABLE  (CCALL,TS6CHE)                  ; check results
   001652' 440000  001645'                      CTABLE  (CJRST,TS6A)                    ; loop till done
   001653' 000000  000000                       CTABLE  (CLAST)                         ; end of table

                                        ; Initialize data pattern selection

   001654' 402 00 0 00 001346*          TS6SET: SETZM   CPAT                            ; clear data pattern
   001655' 201 00 0 00 001000                   MOVEI   ^D512                           ; length of data transfer is
   001656' 336 00 0 00 030037                   SKIPN   USER                            ;    102400 words exec mode
   001657' 201 00 0 00 310000                   MOVEI   ^D102400                        ;    and 512 words user mode
   001660' 202 00 0 00 001375*                  MOVEM   CLEN                            ; save it
   001661' 201 00 0 00 000100                   MOVEI   100                             ; get exec mode start address
   001662' 332 00 0 00 030037                   SKIPE   USER                            ; user mode?
   001663' 201 00 0 00 000110                   MOVEI   110                             ; yes - set up user mode address
   001664' 542 00 0 00 001647'                  HRRM    TS6S                            ; store it
   001665' 263 17 0 00 000000                   RTN                                     ; return

                                        ; Increment data pattern

   001666' 260 17 0 00 001632*          TS6INC: GO      IPACLR                          ; do a 'port clear'
   001667' 350 01 0 00 001654*                  AOS     1,CPAT                          ; increment data pattern
   001670' 307 01 0 00 000114                   CAIG    1,114                           ; done with test yet?
   001671' 350 00 0 17 000000                   AOS     (P)                             ; no - set up proper return
```

```
001672' 332 00 0 00 001352*                 SKIPE   UDEBUG                  ; debug mode?
                                            JRST    [CAIG   1,10            ; yes - limit to only 10 segments
                                                     AOS    (P)
001673' 254 00 0 00 013151'                          JRST   .+1]
001674' 200 01 0 01 000000*                 MOVE    1,PATBUF(1)             ; get data pattern
001675' 202 01 0 00 014011'                 MOVEM   1,SAVDA1                ; save 'correct data'
001676' 135 02 0 00 013154'                 LDB     2,[POINT 10,1,9]        ; get 1st MGC field
001677' 137 02 0 00 013155'                 DPB     2,[POINT 10,T6MM,35]    ; save it
001700' 135 02 0 00 013156'                 LDB     2,[POINT 10,1,19]       ; get 2nd MGC field
001701' 137 02 0 00 013157'                 DPB     2,[POINT 10,T6MM+2,35]  ; save it
001702' 135 02 0 00 013160'                 LDB     2,[POINT 10,1,29]       ; get 3rd MGC field
001703' 137 02 0 00 013161'                 DPB     2,[POINT 10,T6MM+4,35]  ; save it
001704' 135 02 0 00 013162'                 LDB     2,[POINT 6,1,35]        ; get 1st half of 4th MGC field
001705' 137 02 0 00 013163'                 DPB     2,[POINT 6,T6MM+6,31]   ; save it
001706' 316 16 0 00 000000*                 CAMN    MBCN,PORTNI             ; NI port?
001707' 476 00 0 00 000000*                 SETOM   TSLOD1                  ; yes - must reload microcode each time
001710' 316 16 0 00 000000*                 CAMN    MBCN,PORTCI             ; CI port?
001711' 476 00 0 00 000000*                 SETOM   TSLOD2                  ; yes - must reload microcode each time
001712' 201 01 0 00 001717'                 MOVEI   1,T6MM                  ; set up microcode address
001713' 260 17 0 00 001624*                 GO      TLOAD                   ; load/verify it
001714' 255 00 0 00 000000                  JFCL                            ; ignore error
001715' 476 00 0 00 001354*                 SETOM   CSTATF                  ; set check logout status flag
001716' 263 17 0 00 000000                  RTN                             ; return

001717' 000000  010000              T6MM:   MWORD   <ADDR=0,JMAP,J=1,MGC=0,SDO,OR,D=2,SKCN>        ; 0  00-09
001720' 732000  240040
001721' 000100  000000                      MWORD   <CONT,MGC=0,SDO,OR,B=1,D=2,SKCN>               ; 1  10-19
001722' 732000  640340
001723' 000200  000000                      MWORD   <CONT,MGC=0,SDO,OR,B=2,D=2,SKCN>               ; 2  20-29
001724' 732001  240340
001725' 000300  000000                      MWORD   <CONT,MGC=0,SDO,OR,B=3,D=2,SKCN>               ; 3  30-35
001726' 732001  640340
001727' 777777  777777                      -1

                                    ; Check results

001730' 260 17 0 00 001745'         TS6CHE: GO      MA6CHE                  ; get additional data
001731' 200 01 0 00 001440*                 MOVE    1,CEBUFA                ; get word count
001732' 312 01 0 00 001660*                 CAME    1,CLEN                  ; correct?
001733' 474 15 0 00 000000                  SETO    ERFLG,                  ; no - set error flag
001734' 200 01 0 00 014011'                 MOVE    1,SAVDA1                ; get actual data
001735' 312 01 0 00 014012'                 CAME    1,SAVDA2                ; same as correct?
001736' 474 15 0 00 000000                  SETO    ERFLG,                  ; no - set error flag
001737' 350 00 0 17 777772                  AOS     -6(P)                   ; set up error return
001740' 350 00 0 17 777772                  AOS     -6(P)                   ; set up error return
001741' 263 17 0 00 000000                  RTN                             ; return

                                    ; Error messages

001742' 140000  013027'             MA6:    MSG!TXNOT![ASCIZ /Data transfer test error detected/]
001743' 240000  001745'                     CALL!TXNOT!MA6CHE               ; check other results
001744' 000000000000#                       LAST!CALL!TXALL!CCPNT           ; print test data
```

```

                                        ; Check results - first the word count left in the EBUF

 001745' 322 15 0 00 001760'    MA6CHE: JUMPE   ERFLG,MA6CHO                    ; no error yet? - continue
 001746' 200 01 0 00 001731*            MOVE    1,CEBUFA                        ; get word count
 001747' 316 01 0 00 001732*            CAMN    1,CLEN                          ; correct?
 001750' 254 00 0 00 001760'            JRST    MA6CHO                          ; yes - continue
 001751' 037 00 0 00 013061'            TMSGC   <Word count error detected by port:  >
 001752' 037 00 0 00 013071'            TMSGC   <  Correct:>
 001753' 006 10 0 00 001747*            PDEC    10,CLEN
 001754' 037 00 0 00 013074'            TMSG    <.>
 001755' 037 00 0 00 013075'            TMSGC   <  Actual:>
 001756' 006 11 0 00 001746*            PDEC    11,CEBUFA                       ; print it
 001757' 037 00 0 00 013074'            TMSG    <.>

                                ; Then the actual data written

 001760' 400 01 0 00 000000     MA6CHO: SETZ    1,                              ; ensure the port is
 001761' 260 17 0 00 001431*            GO      LDCSR                           ;   stopped
 001762' 201 01 0 00 000300             MOVEI   1,300                           ; get address to get R4
 001763' 242 01 0 00 000001             LSH     1,1                             ; position correctly
 001764' 260 17 0 00 001425*            GO      LDRAR                           ; load RAR register
 001765' 201 01 0 00 000010             MOVEI   1,MPRUN                         ; set up initial CSR data
 001766' 260 17 0 00 001761*            GO      LDCSR                           ; write CSR register
 001767' 201 01 0 00 200000             MOVEI   1,TSTEBF                        ; set up CSR to read EBUF
 001770' 260 17 0 00 001766*            GO      LDCSR                           ; write it
 001771' 260 17 0 00 001432*            GO      RDEBUF                          ; get EBUF data (actual)
 001772' 202 01 0 00 014012'            MOVEM   1,SAVDA2                        ; save it

                                ; Check for data generation error - and print if so

 001773' 332 00 0 00 000015             SKIPE   ERFLG                           ; error yet?
 001774' 316 01 0 00 014011'            CAMN    1,SAVDA1                        ; yes - error?
 001775' 254 00 0 00 002006'            JRST    MA6CH1                          ; no - do data buffer compare
 001776' 037 00 0 00 013164'            TMSGC   <Data pattern not generated correctly:>
 001777' 037 00 0 00 013130'            TMSGC   <  Correct: >
 002000' 200 00 0 00 014011'            MOVE    SAVDA1                          ; get correct data
 002001' 037 13 0 00 000000             PNTHW                                   ; print it
 002002' 037 00 0 00 013133'            TMSGC   <  Actual:  >
 002003' 200 00 0 00 014012'            MOVE    SAVDA2                          ; get actual data
 002004' 037 13 0 00 000000             PNTHW                                   ; print it
 002005' 263 17 0 00 000000             RTN                                     ; return
```

```
                                   ; Check for data buffer compare error

002006' 200 01 0 00 013174'        MA6CH1: MOVE    1,[-1]          ; set to 'read - don't report errors'
002007' 201 02 0 00 001000                 MOVEI   2,^D512         ; get buffer length
002010' 260 17 0 00 000000*                GO      BUFCOM          ; do the compare
002011' 202 01 0 00 014014'                MOVEM   1,SAVNUM#       ; save number of data compare errors
002012' 332 00 0 00 000015                 SKIPE   ERFLG           ; error yet?
002013' 336 00 0 00 000001                 SKIPN   1               ; any words in error?
002014' 263 17 0 00 000000                 RTN                     ; no - return
002015' 201 01 0 00 000001                 MOVEI   1,1             ; set to 'read - report errors'
002016' 201 02 0 00 001000                 MOVEI   2,^D512         ; get buffer length
002017' 260 17 0 00 002010*                GO      BUFCOM          ; do the compare/print errors
002020' 402 00 0 00 000000*                SETZM   CDERR           ; don't report these in CCPNT
002021' 263 17 0 00 000000                 RTN                     ; done - return

                                   ; Microcode - Initialize the word count

                                   ;       Initialize word count (exec mode)

002022' 010001  010000             T6M:    MWORD   <ADDR=100,JMAP,J=101,SD0,AND,B=5,D=2>           ; 100
002023' 742002  400040
002024' 010100  310310                     MWORD   <LDCT,J=31,SD0,OR,B=6,D=2,SKCN,MGC=310>         ; 101
002025' 732003  240300
002026' 010201  020000                     MWORD   <RPCT,J=102,SOB,B=6,OR,D=5>                     ; 102
002027' 335003  000220
002030' 010300  100000                     MWORD   <LDCT,J=10,D=1>                                 ; 103
002031' 001000  000300
002032' 010401  040000                     MWORD   <RPCT,J=104,SOB,B=6,OR,D=6>                     ; 104
002033' 336003  000220
002034' 010500  000000                     MWORD   <JZ>                                            ; 105
002035' 000000  000000

                                   ;       Initialize word count (exec mode)

002036' 011001  110000                     MWORD   <ADDR=110,JMAP,J=111,SD0,AND,B=5,D=2>           ; 110
002037' 742002  400040
002040' 011100  310001                     MWORD   <LDCT,J=31,SD0,OR,B=6,D=2,SKCN,MGC=1>           ; 111
002041' 732003  240300
002042' 011201  120000                     MWORD   <RPCT,J=112,SOB,B=6,OR,D=5>                     ; 112
002043' 335003  000220
002044' 011300  100000                     MWORD   <LDCT,J=10,D=1>                                 ; 113
002045' 001000  000300
002046' 011401  140000                     MWORD   <RPCT,J=114,SOB,B=6,OR,D=6>                     ; 114
002047' 336003  000220
002050' 011500  000000                     MWORD   <JZ>                                            ; 115
002051' 000000  000000

                                   ;       Build the data pattern word

002052' 000000  010000                     MWORD   <ADDR=0,JMAP,J=1,MGC=0,SD0,OR,D=2,SKCN>         ; 0  00-09
002053' 732000  240040
002054' 000100  000000                     MWORD   <CONT,MGC=0,SD0,OR,B=1,D=2,SKCN>                ; 1  10-19
002055' 732000  640340
002056' 000200  000000                     MWORD   <CONT,MGC=0,SD0,OR,B=2,D=2,SKCN>                ; 2  20-29
```

SEQ 1638

SEQ 1651

```
 1711   002057' 732001  240340
 1712   002060' 000300  000000                          MWORD   <CONT,MGC=0,SDO,OR,B=3,D=2,SKCN>        ; 3  30-35
 1713   002061' 732001  640340
 1714   002062' 000400  310000                          MWORD   <LDCT,J=31,D=1>                         ; 4
 1715   002063' 001000  000300
 1716   002064' 000500  050000                          MWORD   <RPCT,J=5,SOB,B=0,OR,D=7>               ; 5
 1717   002065' 337000  000220
 1718   002066' 000600  310000                          MWORD   <LDCT,J=31,D=1>                         ; 6
 1719   002067' 001000  000300
 1720   002070' 000700  070000                          MWORD   <RPCT,J=7,SOB,B=1,OR,D=7>               ; 7
 1721   002071' 337000  400220
 1722   002072' 001000  110000                          MWORD   <LDCT,J=11,D=1>                         ; 10
 1723   002073' 001000  000300
 1724   002074' 001100  110000                          MWORD   <RPCT,J=11,SOB,B=1,OR,D=5>              ; 11
 1725   002075' 335000  400220
 1726   002076' 001200  310000                          MWORD   <LDCT,J=31,D=1>                         ; 12
 1727   002077' 001000  000300
 1728   002100' 001300  130000                          MWORD   <RPCT,J=13,SOB,B=2,OR,D=7>              ; 13
 1729   002101' 337001  000220
 1730   002102' 001400  230000                          MWORD   <LDCT,J=23,D=1>                         ; 14
 1731   002103' 001000  000300
 1732   002104' 001500  150000                          MWORD   <RPCT,J=15,SOB,B=2,OR,D=5>              ; 15
 1733   002105' 335001  000220
 1734   002106' 001600  350000                          MWORD   <LDCT,J=35,D=1>                         ; 16
 1735   002107' 001000  000300
 1736   002110' 001700  170000                          MWORD   <RPCT,J=17,SOB,B=3,OR,D=5>              ; 17
 1737   002111' 335001  400220
 1738   002112' 002000  000000                          MWORD   <CONT,SOA,OR,D=2,B=4>                   ; 20
 1739   002113' 432002  000340
 1740   002114' 002100  000000                          MWORD   <CONT,SAB,OR,D=2,A=1,B=4>               ; 21
 1741   002115' 132012  000340
 1742   002116' 002200  000000                          MWORD   <CONT,SAB,OR,D=2,A=2,B=4>               ; 22
 1743   002117' 132022  000340
 1744   002120' 002300  000000                          MWORD   <CONT,SAB,OR,D=2,A=3,B=4>               ; 23
 1745   002121' 132032  000340
 1746
 1747                                           ;       Write all the words specified by the CCW list
 1748
 1749   002122' 002405  000220                          MWORD   <CJS,J=500,SELC,MGC=220>                ; 24
 1750   002123' 000000  004020
 1751   002124' 002500  350000                          MWORD   <CJP,J=35,CENA,CCCE>                    ; 25
 1752   002125' 000400  160060
 1753   002126' 002600  350000                          MWORD   <CJP,J=35,CENA,CCCP>                    ; 26
 1754   002127' 000400  140060
 1755   002130' 002700  310000                          MWORD   <CJP,J=31,CENA,CCCA>                    ; 27
 1756   002131' 000400  000060
 1757   002132' 003000  250000                          MWORD   <JMAP,J=25>                             ; 30
 1758   002133' 000000  000040
 1759   002134' 003100  000000                          MWORD   <CONT>                                  ; 31
 1760   002135' 000000  000340
 1761   002136' 003200  002040                          MWORD   <CONT,SOA,A=4,D=1,OENA,SELF,MGC=40>     ; 32
 1762   002137' 401040  003340
 1763   002140' 003300  000002                          MWORD   <CONT,SOB,PLUS,B=5,D=2,CRY,SELC,MGC=2>  ; 33
 1764   002141' 302002  404740
 1765   002142' 003400  360000                          MWORD   <CJP,J=36,SOB,SMIN,B=6,D=2,CENA,CCFZ>   ; 34
```

```
002143' 312403  020060
002144' 003500  250000                        MWORD   <JMAP,J=25>                                     ; 35
002145' 000000  000040

                                      ;       All done -  set STOP and STORE (after a short delay to wait
                                      ;       for channel to finish up the last data transfer).

002146' 003600  100000                        MWORD   <LDCT,J=10>                                     ; 36
002147' 000000  000300
002150' 003700  370000                        MWORD   <RPCT,J=37>                                     ; 37
002151' 000000  000220
002152' 004000  000140                        MWORD   <CONT,SELC,MGC=140>                             ; 40
002153' 000000  004340

                                      ;       Finally, write bit 'RspQAV' in CSR to indicate we are done,
                                      ;       and write word count to EBUF

002154' 004100  310200                        MWORD   <LDCT,J=31,SDO,OR,B=17,D=2,SKCN,MGC=200>        ; 41
002155' 732007  640300
002156' 004200  420000                        MWORD   <RPCT,J=42,SOB,OR,B=17,D=5>                     ; 42
002157' 335007  400220
002160' 004300  002004                        MWORD   <CONT,SOA,A=17,OR,D=1,OENA,SELE,MGC=4>          ; 43
002161' 431170  005340
002162' 004400  460040                        MWORD   <CJP,J=46,CENA,CCGC,SELE,MGC=40>                ; 44
002163' 000400  015060
002164' 004500  440000                        MWORD   <JMAP,J=44>                                     ; 45
002165' 000000  000040
002166' 004600  000200                        MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>              ; 46
002167' 431010  005340
002170' 004700  002004                        MWORD   <CONT,SOA,OR,A=5,D=1,OENA,SELE,MGC=4>           ; 47
002171' 431050  005340
002172' 005000  500000                        MWORD   <JMAP,J=50>                                     ; 50
002173' 000000  000040

                                      ;       Routine to read R4 and place in EBUF

002174' 030003  012004                        MWORD   <ADDR=300,JMAP,J=301,SOA,A=4,OR,OENA,D=1,SELE,MGC=4>
002175' 431040  005040
002176' 030103  010000                        MWORD   <JMAP,J=301>                                    ; 301
002177' 000000  000040

                                      ;       Delay

002200' 050000  100000                        MWORD   <ADDR=500,LDCT,J=10,D=1>                        ; 500
002201' 001000  000300
002202' 050105  010000                        MWORD   <RPCT,J=501,D=1>                                ; 501
002203' 001000  000220
002204' 050200  000000                        MWORD   <CRTN>                                          ; 502
002205' 000000  000240

                                      ;       Initialization

002206' 100000  000020                        MWORD   <ADDR=1000,JZ,SELM,MGC=20>                      ; 1000
002207' 000000  002000

```

```
                                      ;       Initialization step
```

```
 1821                                   ;       Initialization step
 1822
 1823  002210' 500050  010100                   MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>            ; 5000
 1824  002211' 000000  004040
 1825  002212' 500150  010020                   MWORD   <JMAP,J=5001,SELM,MGC=20>                       ; 5001
 1826  002213' 000000  002040
 1827  002214' 777777  777777                   -1
```

```
 1828
```

```
                                        ;#*********************************************************************
                                        ;* Test 7 - CBUS to EBUS Data Transfer
                                        ;
                                        ; Description:  This test does data transfers over the  CBUS  and
                                        ;               EBUS concurrently.  Data is read by the port over
                                        ;               the CBUS, then transfered via an IOP deposit over
                                        ;               the EBUS into the 2nd half of the  buffer.   Then
                                        ;               a data compare of the  data  buffer  is  done  to
                                        ;               verify the result.
                                        ;
                                        ; Procedure:    KL> Port Clear
                                        ;               KL> Set up a CCW list to transfer 1 word
                                        ;               KL> Start up port
                                        ;               UC> Read word from CBUS and save as buffer address+256
                                        ;               KL> Set up a CCW list to transfer 256 words
                                        ;               KL> Start up port
                                        ;               UC> Read word from CBUS and write over EBUS into 2nd
                                        ;                   half of buffer
                                        ;               KL> When done, verify data transfer completed ok
                                        ;
                                        ;               Repeat with data patterns 116-125
                                        ;
                                        ; Failure:      ---
                                        ;#*********************************************************************

                                        ; Test data

002215' 254 00 0 00 002225'             TSTC7:  JRST    TG7                     ; go start test
002216' 100604  000007                          CBUS!NDMP!NDCB!ZCBUS!7          ; test mask
002217' 002334' 013175'                         T7M,,[ASCIZ ^CBUS to EBUS Data Transfer^]
002220' 013203' 013051'                         [EXP E1,E4,C1,C14,MLAST!C17],,[EXP C12,C5,C8,C4,MLAST!C2]
002221' 000000  002517'                         TSTC10                          ; failure test table
002222' 000000000000#                           TSTC40                          ;   ...
002223' 000000000000#                           TSTC41
002224' 777777  777777                          -1

                                        ; Start test

002225' 201 00 0 00 000000'             TG7:    MOVEI   Z9                      ; get address of module start
002226' 260 17 0 00 001622*                     GO      TRACE                   ; handle trace output
002227' 201 01 0 00 002334'                     MOVEI   1,T7M                   ; set up microcode address
002230' 260 17 0 00 001713*                     GO      TLOAD                   ; load/verify it
002231' 263 17 0 00 000000                      RTN                             ; failed - exit test

                                        ; Initialization

002232' 400 15 0 00 000000              TL7:    SETZ    ERFLG,                  ; clear error flag
002233' 201 06 0 00 002301'                     MOVEI   6,TS7                   ; get test table address
002234' 402 00 0 00 001630*                     SETZM   TSTSUB                  ; initialize subtest number
002235' 400 15 0 00 000000                      SETZ    ERFLG,                  ; clear error flag
```

```

                                   ; Initialize the data transfer

002236' 260 17 0 00 001666*                  GO      IPACLR                  ; clear port
002237' 201 01 0 00 004000                   MOVEI   1,4000                  ; get start address
002240' 242 01 0 00 000001                   LSH     1,1                     ; position correctly
002241' 260 17 0 00 001764*                  GO      LDRAR                   ; load the register
002242' 201 01 0 00 000010                   MOVEI   1,MPRUN                 ; get 'MPRUN' bit
002243' 260 17 0 00 001770*                  GO      LDCSR                   ; start up the port

                                   ; Set up channel

002244' 201 01 0 00 000000#                  MOVEI   1,BUFF+^D256            ; get buffer address
002245' 336 00 0 00 001672*                  SKIPN   UDEBUG                  ; debug mode?
002246' 257 01 0 01 000000                   MAP     1,(1)                   ; no - map to physical
002247' 621 01 0 00 777000                   TLZ     1,777000                ; get rid of extraneous bits
002250' 202 01 0 00 001116*                  MOVEM   1,BUFF                  ; save in buffer
002251' 200 01 0 00 013210'                  MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
002252' 260 17 0 00 000000*                  GO      CHINIT                  ; initialize software
002253' 551 01 0 00 002250*                  HRRZI   1,BUFF                  ; buffer address
002254' 201 02 0 00 000001                   MOVEI   2,1                     ; word count
002255' 400 03 0 00 000000                   SETZ    3,                      ; clear zero fill specifier
002256' 260 17 0 00 000000*                  GO      GENCCW                  ; generate a CCW list

                                   ; Do the data transfer

002257' 260 17 0 00 002236*                  GO      IPACLR                  ; clear the port
002260' 201 01 0 00 004002                   MOVEI   1,4002                  ; get start address
002261' 242 01 0 00 000001                   LSH     1,1                     ; position correctly
002262' 260 17 0 00 002241*                  GO      LDRAR                   ; load the register
002263' 201 01 0 00 000010                   MOVEI   1,MPRUN                 ; get 'MPRUN' bit
002264' 260 17 0 00 002243*                  GO      LDCSR                   ; start up the port
002265' 201 00 0 00 000010                   MOVEI   10                      ; set up 10ms wait
002266' 260 17 0 00 000000*                  GO      ODELAY                  ; wait a bit

                                   ; Loop on test execute table entries

002267' 260 17 0 00 002257*        TA7:      GO      IPACLR                  ; clear port
002270' 260 17 0 00 001633*        TB7:      GO      CEXEC                   ; execute table entry
002271' 254 00 0 00 002300'                   JRST    TX7                    ; end of test execute table
002272' 254 00 0 00 002270'                   JRST    TB7                    ; keep looping after call
002273' 474 15 0 00 000000                    SETO    ERFLG,                 ; error occurred

                                   ; Handle error printouts and scope looping

002274' 027 00 0 00 002321'                  SCOPER  MA7                     ; print error message
002275' 254 00 0 00 002232'                  JRST    TL7                     ; loop on error
002276' 254 00 0 00 002300'                  JRST    TX7                     ; altmode exit
002277' 322 15 0 00 002267'                  JUMPE   ERFLG,TA7               ; do next test execute table entry

                                   ; End of test

002300' 263 17 0 00 000000         TX7:      RTN                             ; return
```

```

                                           ; Test Execute Entries

002301' 240000  002311'                   TS7:    CTABLE  (CCALL,TS7SE)           ; set up initial data pattern
002302' 300000  002314'                   TS7A:   CTABLE  (CCALLC,TS7IN)          ; increment data pattern
002303' 140000  001000                            CTABLE  (CSETRW,0,^D512)        ; set up read/write (512 words)
002304' 200000  001000                            CTABLE  (CSTART,1000)           ; start up port
002305' 340100  000000                            CTABLE  (CWAIT,100,0)           ; check if completed
002306' 400000  000000                            CTABLE  (CCOMP)                 ; compare buffer
002307' 440000  002302'                           CTABLE  (CJRST,TS7A)            ; loop till done
002310' 000000  000000                            CTABLE  (CLAST)                 ; end of table

                                           ; Initialize data pattern selection

002311' 201 00 0 00 000115                TS7SE:  MOVEI   115                     ; get initial pattern
002312' 202 00 0 00 001667*                       MOVEM   CPAT                    ; save data pattern
002313' 263 17 0 00 000000                        RTN                             ; return

                                           ; Increment data pattern

002314' 350 01 0 00 002312*               TS7IN:  AOS     1,CPAT                  ; increment data pattern
002315' 260 17 0 00 002267*                       GO      IPACLR                  ; do a 'port clear'
002316' 307 01 0 00 000125                        CAIG    1,125                   ; done with test yet?
002317' 350 00 0 17 000000                        AOS     (P)                     ; no - set up proper return
002320' 263 17 0 00 000000                        RTN                             ; return

                                           ; Error messages

002321' 140000  013027'                   MA7:    MSG!TXNOT![ASCIZ /Data transfer test error detected/]
002322' 240000  002324'                           CALL!TXNOT!MA7CHE               ; check for data compare error
002323' 000000000000#                             LAST!CALL!TXALL!CCPNT           ; print test data

002324' 336 00 0 00 002020*               MA7CHE: SKIPN   CDERR                   ; data compare error?
002325' 263 17 0 00 000000                        RTN                             ; no - return
002326' 201 01 0 00 000003                        MOVEI   1,3                     ; get transfer type
002327' 201 02 0 00 001000                        MOVEI   2,^D512                 ; get transfer length
002330' 200 14 0 00 002314*                       MOVE    PAT,CPAT                ; get data pattern
002331' 260 17 0 00 002017*                       GO      BUFCOM                  ; compare the data
002332' 402 00 0 00 002324*                       SETZM   CDERR                   ; don't report these in CCPNT
002333' 263 17 0 00 000000                        RTN                             ; return

                                           ; Microcode - Read a word into R3 and write to 2nd half of buffer

002334' 000000  020000                    T7M:    MWORD   <ADDR=0,CJP,J=2,CENA,CCCA>                      ; 0
002335' 000400  000060
002336' 000100  000000                            MWORD   <JZ>                                            ; 1
002337' 000000  000000
002340' 000200  000004                            MWORD   <CONT,SELC,MGC=4>                               ; 2
002341' 000000  004340
002342' 000300  000200                            MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>              ; 3
002343' 732001  402340

002344' 000400  002004                            MWORD   <CONT,SAB,OR,D=1,A=15,B=16,OENA,SELE,MGC=4>     ; 4
002345' 131157  005340
002346' 000500  000001                            MWORD   <CONT,SOB,PLUS,B=17,D=2,CRY,SELE,MGC=1>         ; 5
```

```
002347' 302007  405740                  MWORD   <CJP,J=10,CENA,CCER>                            ; 6
002350' 000600  100000
002351' 000400  100060                  MWORD   <JMAP,J=6>                                      ; 7
002352' 000700  060000
002353' 000000  000040                  MWORD   <CONT,SOA,OR,A=3,D=1,SELE,MGC=4,OENA>           ; 10
002354' 001000  002004
002355' 431030  005340                  MWORD   <CJS,J=100,SOB,PLUS,B=15,CRY,D=2,SELE,MGC=20>   ; 11
002356' 001101  000020
002357' 302006  405420

002360' 001200  000000                  MWORD   <RPCT,J=0>                                      ; 12
002361' 000000  000220

                                ;       All done -  set STOP and STORE

002362' 001300  000140                  MWORD   <CONT,SELC,MGC=140>                             ; 13
002363' 000000  004340

                                ;       Finally, write bit 'RspQAV' in CSR to indicate we are done,
                                ;       and write word count to EBUF

002364' 001400  310200                  MWORD   <LDCT,J=31,SDO,OR,B=17,D=2,SKCN,MGC=200>        ; 14
002365' 732007  640300
002366' 001500  150000                  MWORD   <RPCT,J=15,SOB,OR,B=17,D=5>                     ; 15
002367' 335007  400220
002370' 001600  002004                  MWORD   <CONT,SOA,A=17,OR,D=1,OENA,SELE,MGC=4>          ; 16
002371' 431170  005340
002372' 001700  210040                  MWORD   <CJP,J=21,CENA,CCGC,SELE,MGC=40>                ; 17
002373' 000400  015060
002374' 002000  170000                  MWORD   <JMAP,J=17>                                     ; 20
002375' 000000  000040
002376' 002100  000200                  MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>              ; 21
002377' 431010  005340
002400' 002200  002004                  MWORD   <CONT,SOA,OR,A=5,D=1,OENA,SELE,MGC=4>           ; 22
002401' 431050  005340
002402' 002300  230000                  MWORD   <JMAP,J=23>                                     ; 23
002403' 000000  000040

                                ;       Delay

002404' 010000  000000                  MWORD   <ADDR=100,CONT>                                 ; 100
002405' 000000  000340
002406' 010100  000000                  MWORD   <CONT>                                          ; 101
002407' 000000  000340
002410' 010200  000000                  MWORD   <CONT>                                          ; 102
002411' 000000  000340
002412' 010300  000000                  MWORD   <CONT>                                          ; 103
002413' 000000  000340
002414' 010400  000000                  MWORD   <CONT>                                          ; 104
002415' 000000  000340
002416' 010500  000000                  MWORD   <CONT>                                          ; 105
002417' 000000  000340
002420' 010600  000000                  MWORD   <CONT>                                          ; 106
002421' 000000  000340
002422' 010700  000000                  MWORD   <CONT>                                          ; 107
```

```
2042  002423' 000000  000340
2043  002424' 011000  000000                        MWORD   <CONT>                                                  ; 110
2044  002425' 000000  000340
2045  002426' 011100  000000                        MWORD   <CONT>                                                  ; 111
2046  002427' 000000  000340
2047  002430' 011200  000000                        MWORD   <CONT>                                                  ; 112
2048  002431' 000000  000340
2049  002432' 011300  000000                        MWORD   <CONT>                                                  ; 113
2050  002433' 000000  000340
2051  002434' 011400  000000                        MWORD   <CONT>                                                  ; 114
2052  002435' 000000  000340
2053  002436' 011500  000000                        MWORD   <CONT>                                                  ; 115
2054  002437' 000000  000340
2055  002440' 011600  000000                        MWORD   <CONT>                                                  ; 116
2056  002441' 000000  000340
2057  002442' 011700  000000                        MWORD   <CONT>                                                  ; 117
2058  002443' 000000  000340
2059  002444' 012000  000000                        MWORD   <CONT>                                                  ; 120
2060  002445' 000000  000340
2061  002446' 012100  000000                        MWORD   <CONT>                                                  ; 121
2062  002447' 000000  000340
2063  002450' 012200  000000                        MWORD   <CONT>                                                  ; 122
2064  002451' 000000  000340
2065  002452' 012300  000000                        MWORD   <CRTN>                                                  ; 123
2066  002453' 000000  000240
2067
2068                                        ;       Initialization
2069
2070  002454' 100010  010020                        MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>                     ; 1000
2071  002455' 000000  002040
2072  002456' 100103  770004                        MWORD   <LDCT,J=377,SOA,A=14,B=15,OR,D=2,SELC,MGC=4>            ; 1001
2073  002457' 432146  404300
2074  002460' 100200  000200                        MWORD   <JZ,SELC,MGC=200>                                       ; 1002
2075  002461' 000000  004000
2076
2077                                        ;       Initialization step to read buffer address into R14,R15 and
2078                                        ;       put IOP word (left half) into R16
2079
2080  002462' 400040  010100                        MWORD   <ADDR=4000,JMAP,J=4001,SELC,MGC=100>                    ; 4000
2081  002463' 000000  004040
2082  002464' 400140  010020                        MWORD   <JMAP,J=4001,SELM,MGC=20>                               ; 4001
2083  002465' 000000  002040
2084  002466' 400240  030200                        MWORD   <JMAP,J=4003,SELC,MGC=200>                              ; 4002
2085  002467' 000000  004040
2086  002470' 400340  050000                        MWORD   <CJP,J=4005,CENA,CCCA>                                  ; 4003
2087  002471' 000400  000060
2088  002472' 400440  030000                        MWORD   <JMAP,J=4003>                                           ; 4004
2089  002473' 000000  000040
2090  002474' 400500  000004                        MWORD   <CONT,SELC,MGC=4>                                       ; 4005
2091  002475' 000000  004340
2092  002476' 400600  000200                        MWORD   <CONT,SDO,OR,B=15,D=2,SELM,MGC=200>                     ; 4006
2093  002477' 732006  402340
2094  002500' 400700  000000                        MWORD   <CONT,SOA,OR,A=15,B=14,D=2>                             ; 4007
2095  002501' 432156  000340
2096  002502' 401000  320450                        MWORD   <LDCT,J=32,SDO,OR,B=16,D=2,SKCN,MGC=450>                ; 4010
```

```
002503' 732007  240300                    MWORD   <RPCT,J=4011,SOB,B=16,OR,D=7>              ; 4011
002504' 401140  110000
002505' 337007  000220                    MWORD   <CONT,SELC,MGC=140>                        ; 4012
002506' 401200  000140
002507' 000000  004340                    MWORD   <JMAP,J=4013,SOB,B=17,AND,D=2>             ; 4013
002510' 401340  130000
002511' 342007  400040

                                  ;       Initialization step

002512' 500050  010100                    MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>       ; 5000
002513' 000000  004040
002514' 500150  010020                    MWORD   <JMAP,J=5001,SELM,MGC=20>                  ; 5001
002515' 000000  002040
002516' 777777  777777                    -1
```

```
                                        ;#*********************************************************************
                                        ;* Test 10 - EBUS to CBUS Data Transfer
                                        ;
                                        ; Description:  This test does data transfers over the  CBUS  and
                                        ;               EBUS concurrently.  Data is read by the port over
                                        ;               the EBUS via an IOP examine, then transfered over
                                        ;               the CBUS into the 1st half of the  buffer.   Then
                                        ;               a data compare of the  data  buffer  is  done  to
                                        ;               verify the result.
                                        ;
                                        ; Procedure:    KL> Port Clear
                                        ;               KL> Set up a CCW list to transfer 1 word
                                        ;               KL> Start up port
                                        ;               UC> Read word from CBUS and save as buffer address+256
                                        ;               KL> Set up a CCW list to transfer 256 words
                                        ;               KL> Start up port
                                        ;               UC> Read word from EBUS and write over CBUS into 1st
                                        ;                   half of buffer
                                        ;               KL> When done, verify data transfer completed ok
                                        ;
                                        ;               Repeat with data patterns 116-125
                                        ;
                                        ; Failure:      ---
                                        ;#*********************************************************************

                                        ; Test data

002517' 254 00 0 00 002526'             TSTC10: JRST    TG10                    ; go start test
002520' 100604  000010                          CBUS!NDMP!NDCB!ZCBUS!10         ; test mask
002521' 002645' 013211'                         T10M,,[ASCIZ ^EBUS to CBUS Data Transfer^]
002522' 013217' 013051'                         [EXP E1,E4,C1,C15,MLAST!C16],,[EXP C12,C5,C8,C4,MLAST!C2]
002523' 000000000000#                           TSTC40                          ; failure test table
002524' 000000000000#                           TSTC41                          ;   ...
002525' 777777  777777                          -1

                                        ; Start test

002526' 201 00 0 00 000000'             TG10:   MOVEI   Z9                      ; get address of module start
002527' 260 17 0 00 002226*                     GO      TRACE                   ; handle trace output
002530' 201 01 0 00 002645'                     MOVEI   1,T10M                  ; set up microcode address
002531' 260 17 0 00 002230*                     GO      TLOAD                   ; load/verify it
002532' 263 17 0 00 000000                      RTN                             ; failed - exit test

                                        ; Initialization

002533' 400 15 0 00 000000              TL10:   SETZ    ERFLG,                  ; clear error flag
002534' 201 06 0 00 002602'                     MOVEI   6,TS10                  ; get test table address
002535' 402 00 0 00 002234*                     SETZM   TSTSUB                  ; initialize subtest number
002536' 400 15 0 00 000000                      SETZ    ERFLG,                  ; clear error flag
```

```


				; Initialize the data transfer

 002537' 260 17 0 00 002315*		GO	IPACLR			; clear port
 002540' 201 01 0 00 004000		MOVEI	1,4000			; get start address
 002541' 242 01 0 00 000001		LSH	1,1			; position correctly
 002542' 260 17 0 00 002262*		GO	LDRAR			; load the register
 002543' 201 01 0 00 000010		MOVEI	1,MPRUN			; get 'MPRUN' bit
 002544' 260 17 0 00 002264*		GO	LDCSR			; start up the port

				; Set up channel

 002545' 201 01 0 00 000000#		MOVEI	1,BUFF+^D256		; get buffer address
 002546' 336 00 0 00 002245*		SKIPN	UDEBUG			; debug mode?
 002547' 257 01 0 01 000000		MAP	1,(1)			; no - map to physical
 002550' 621 01 0 00 777000		TLZ	1,777000		; get rid of extraneous bits
 002551' 202 01 0 00 002253*		MOVEM	1,BUFF			; save in buffer
 002552' 200 01 0 00 013210'		MOVE	1,[CBUFSZ,,CBUF]	; get pointer to control buffer
 002553' 260 17 0 00 002252*		GO	CHINIT			; initialize software
 002554' 551 01 0 00 002551*		HRRZI	1,BUFF			; buffer address
 002555' 201 02 0 00 000001		MOVEI	2,1			; word count
 002556' 400 03 0 00 000000		SETZ	3,			; clear zero fill specifier
 002557' 260 17 0 00 002256*		GO	GENCCW			; generate a CCW list

				; Do the data transfer

 002560' 260 17 0 00 002537*		GO	IPACLR			; clear the port
 002561' 201 01 0 00 004002		MOVEI	1,4002			; get start address
 002562' 242 01 0 00 000001		LSH	1,1			; position correctly
 002563' 260 17 0 00 002542*		GO	LDRAR			; load the register
 002564' 201 01 0 00 000010		MOVEI	1,MPRUN			; get 'MPRUN' bit
 002565' 260 17 0 00 002544*		GO	LDCSR			; start up the port
 002566' 201 00 0 00 000010		MOVEI	10			; set up 10ms wait
 002567' 260 17 0 00 002266*		GO	ODELAY			; wait a bit

				; Loop on test execute table entries

 002570' 260 17 0 00 002560*	TA10:	GO	IPACLR			; clear port
 002571' 260 17 0 00 002270*	TB10:	GO	CEXEC			; execute table entry
 002572' 254 00 0 00 002601'		 JRST	TX10			; end of test execute table
 002573' 254 00 0 00 002571'		 JRST	TB10			; keep looping after call
 002574' 474 15 0 00 000000		 SETO	ERFLG,			; error occurred

				; Handle error printouts and scope looping

 002575' 027 00 0 00 002632'		SCOPER	MA10			; print error message
 002576' 254 00 0 00 002533'		JRST	TL10			; loop on error
 002577' 254 00 0 00 002601'		JRST	TX10			; altmode exit
 002600' 322 15 0 00 002570'		JUMPE	ERFLG,TA10		; do next test execute table entry

				; End of test

 002601' 263 17 0 00 000000	TX10:	RTN				; return
```

```

                                       ; Test Execute Entries

002602' 240000  002613'                TS10:   CTABLE  (CCALL,TS10SE)          ; set up initial data pattern
002603' 300000  002616'                TS10A:  CTABLE  (CCALLC,TS10IN)         ; increment data pattern
002604' 140000  001000                         CTABLE  (CSETRW,0,^D512)        ; set up read/write (512 words)
002605' 240000  002623'                        CTABLE  (CCALL,TS10SW)          ; swap buffer halves
002606' 200000  001000                         CTABLE  (CSTART,1000)           ; start up port
002607' 340100  000000                         CTABLE  (CWAIT,100,0)           ; check if completed
002610' 400000  000000                         CTABLE  (CCOMP)                 ; compare buffer
002611' 440000  002603'                        CTABLE  (CJRST,TS10A)           ; loop till done
002612' 000000  000000                         CTABLE  (CLAST)                 ; end of table

                                       ; Initialize data pattern selection

002613' 201 00 0 00 000115             TS10SE: MOVEI   115                     ; get initial pattern
002614' 202 00 0 00 002330*                    MOVEM   CPAT                    ; save data pattern
002615' 263 17 0 00 000000                     RTN                             ; return

                                       ; Increment data pattern

002616' 350 01 0 00 002614*            TS10IN: AOS     1,CPAT                  ; increment data pattern
002617' 260 17 0 00 002570*                    GO      IPACLR                  ; do a 'port clear'
002620' 307 01 0 00 000126                     CAIG    1,126                   ; done with test yet?
002621' 350 00 0 17 000000                     AOS     (P)                     ; no - set up proper return
002622' 263 17 0 00 000000                     RTN                             ; return

                                       ; Swap buffer halves

002623' 200 00 0 00 013224'            TS10SW: MOVE    [BUFF,,BUFF+400]        ; move first 256 words of buffer
002624' 251 00 0 00 000000#                    BLT     BUFF+777                ;   into 2nd half of buffer
002625' 200 00 0 00 013225'                    MOVE    [400000,,1]             ; initialize first half of buffer
002626' 202 00 0 00 002554*                    MOVEM   BUFF                    ;   now
002627' 200 00 0 00 013226'                    MOVE    [BUFF,,BUFF+1]          ;   ...
002630' 251 00 0 00 000000#                    BLT     BUFF+377                ;   ...
002631' 263 17 0 00 000000                     RTN                             ; return

                                       ; Error messages

002632' 140000  013027'                MA10:   MSG!TXNOT![ASCIZ /Data transfer test error detected/]
002633' 240000  002635'                        CALL!TXNOT!MA10CH               ; check for data compare error
002634' 000000000000#                          LAST!CALL!TXALL!CCPNT           ; print test data

002635' 336 00 0 00 002332*            MA10CH: SKIPN   CDERR                   ; data compare error?
002636' 263 17 0 00 000000                     RTN                             ; no - return
002637' 201 01 0 00 000003                     MOVEI   1,3                     ; get transfer type
002640' 201 02 0 00 000400                     MOVEI   2,^D256                 ; get transfer length
002641' 200 14 0 00 002616*                    MOVE    PAT,CPAT                ; get data pattern
002642' 260 17 0 00 002331*                    GO      BUFCOM                  ; compare the data
002643' 402 00 0 00 002635*                    SETZM   CDERR                   ; don't report these in CCPNT
002644' 263 17 0 00 000000                     RTN                             ; return
```

```
                                        ; Microcode - Read a word into R3 and write to 1st half of buffer

002645' 000003  770000                  T10M:   MWORD   <ADDR=0,LDCT,J=377,SOB,B=17,AND,D=2>           ; 0
002646' 342007  400300

002647' 000100  002004                          MWORD   <CONT,SAB,OR,D=1,A=15,B=16,OENA,SELE,MGC=4>    ; 1
002650' 131157  005340
002651' 000200  000001                          MWORD   <CONT,SOB,PLUS,B=15,D=2,CRY,SELE,MGC=1>        ; 2
002652' 302006  405740
002653' 000300  050000                          MWORD   <CJP,J=5,CENA,CCER>                            ; 3
002654' 000400  100060
002655' 000400  030000                          MWORD   <JMAP,J=3>                                     ; 4
002656' 000000  000040
002657' 000500  000010                          MWORD   <CONT,SDO,OR,B=3,D=2,SELE,MGC=10>              ; 5
002660' 732001  405340

002661' 000600  100000                          MWORD   <CJP,J=10,CENA,CCCA>                           ; 6
002662' 000400  000060
002663' 000700  060000                          MWORD   <JMAP,J=6>                                     ; 7
002664' 000000  000040
002665' 001000  002040                          MWORD   <CONT,SOB,B=3,OR,D=1,OENA,SELF,MGC=40>         ; 10
002666' 331001  403340
002667' 001101  000002                          MWORD   <CJS,J=100,D=1,SELC,MGC=2>                     ; 11
002670' 001000  004020

002671' 001200  010000                          MWORD   <RPCT,J=1,SOB,B=17,PLUS,CRY,D=2>               ; 12
002672' 302007  400620

                                        ;       All done -  set STOP and STORE (after a short delay to wait
                                        ;       for channel to finish up the last data transfer).

002673' 001300  100000                          MWORD   <LDCT,J=10>                                    ; 13
002674' 000000  000300
002675' 001400  140000                          MWORD   <RPCT,J=14>                                    ; 14
002676' 000000  000220
002677' 001500  000140                          MWORD   <CONT,SELC,MGC=140>                            ; 15
002700' 000000  004340

                                        ;       Finally, write bit 'RspQAV' in CSR to indicate we are done,
                                        ;       and write word count to EBUF

002701' 001600  310200                          MWORD   <LDCT,J=31,SDO,OR,B=17,D=2,SKCN,MGC=200>       ; 16
002702' 732007  640300
002703' 001700  170000                          MWORD   <RPCT,J=17,SOB,OR,B=17,D=5>                    ; 17
002704' 335007  400220
002705' 002000  002004                          MWORD   <CONT,SOA,A=17,OR,D=1,OENA,SELE,MGC=4>         ; 20
002706' 431170  005340
002707' 002100  230040                          MWORD   <CJP,J=23,CENA,CCGC,SELE,MGC=40>               ; 21
002710' 000400  015060
002711' 002200  210000                          MWORD   <JMAP,J=21>                                    ; 22
002712' 000000  000040
002713' 002300  000200                          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>             ; 23
002714' 431010  005340
002715' 002400  002004                          MWORD   <CONT,SOA,OR,A=5,D=1,OENA,SELE,MGC=4>          ; 24
```

```
002716' 431050  005340          MWORD   <JMAP,J=25>                             ; 25
002717' 002500  250000
002720' 000000  000040

                        ;       Delay

002721' 010000  000000          MWORD   <ADDR=100,CONT>                         ; 100
002722' 000000  000340
002723' 010100  000000          MWORD   <CONT>                                  ; 101
002724' 000000  000340
002725' 010200  000000          MWORD   <CONT>                                  ; 102
002726' 000000  000340
002727' 010300  000000          MWORD   <CONT>                                  ; 103
002730' 000000  000340
002731' 010400  000000          MWORD   <CONT>                                  ; 104
002732' 000000  000340
002733' 010500  000000          MWORD   <CONT>                                  ; 105
002734' 000000  000340
002735' 010600  000000          MWORD   <CONT>                                  ; 106
002736' 000000  000340
002737' 010700  000000          MWORD   <CONT>                                  ; 107
002740' 000000  000340
002741' 011000  000000          MWORD   <CONT>                                  ; 110
002742' 000000  000340
002743' 011100  000000          MWORD   <CONT>                                  ; 111
002744' 000000  000340
002745' 011200  000000          MWORD   <CONT>                                  ; 112
002746' 000000  000340
002747' 011300  000000          MWORD   <CONT>                                  ; 113
002750' 000000  000340
002751' 011400  000000          MWORD   <CONT>                                  ; 114
002752' 000000  000340
002753' 011500  000000          MWORD   <CONT>                                  ; 115
002754' 000000  000340
002755' 011600  000000          MWORD   <CONT>                                  ; 116
002756' 000000  000340
002757' 011700  000000          MWORD   <CONT>                                  ; 117
002760' 000000  000340
002761' 012000  000000          MWORD   <CONT>                                  ; 120
002762' 000000  000340
002763' 012100  000000          MWORD   <CONT>                                  ; 121
002764' 000000  000340
002765' 012200  000000          MWORD   <CONT>                                  ; 122
002766' 000000  000340
002767' 012300  000000          MWORD   <CRTN>                                  ; 123
002770' 000000  000240

                        ;       Initialization

002771' 100010  010020          MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>     ; 1000
002772' 000000  002040
002773' 100100  000004          MWORD   <CONT,SOA,A=14,B=15,OR,D=2,SELC,MGC=4>  ; 1001
002774' 432146  404340
002775' 100200  000220          MWORD   <JZ,SELC,MGC=220>                       ; 1002
002776' 000000  004000
```

```

                                  ;       Initialization step to read buffer address into R14,R15 and
                                  ;       put IOP word (left half) into R16

002777' 400040  010100                    MWORD   <ADDR=4000,JMAP,J=4001,SELC,MGC=100>            ; 4000
003000' 000000  004040
003001' 400140  010020                    MWORD   <JMAP,J=4001,SELM,MGC=20>                       ; 4001
003002' 000000  002040
003003' 400240  030200                    MWORD   <JMAP,J=4003,SELC,MGC=200>                      ; 4002
003004' 000000  004040
003005' 400340  050000                    MWORD   <CJP,J=4005,CENA,CCCN>                          ; 4003
003006' 000400  000060
003007' 400440  030000                    MWORD   <JMAP,J=4003>                                   ; 4004
003010' 000000  000040
003011' 400500  000004                    MWORD   <CONT,SELC,MGC=4>                               ; 4005
003012' 000000  004340
003013' 400600  000200                    MWORD   <CONT,SDO,OR,B=15,D=2,SELM,MGC=200>             ; 4006
003014' 732006  402340
003015' 400700  000000                    MWORD   <CONT,SOA,OR,A=15,B=14,D=2>                     ; 4007
003016' 432156  000340
003017' 401000  320440                    MWORD   <LDCT,J=32,SDO,OR,B=16,D=2,SKCN,MGC=440>        ; 4010
003020' 732007  240300
003021' 401140  110000                    MWORD   <RPCT,J=4011,SOB,B=16,OR,D=7>                   ; 4011
003022' 337007  000220
003023' 401200  000140                    MWORD   <CONT,SELC,MGC=140>                             ; 4012
003024' 000000  004340
003025' 401340  130000                    MWORD   <JMAP,J=4013,SOB,B=17,AND,D=2>                  ; 4013
003026' 342007  400040

                                  ;       Initialization step

003027' 500050  010100                    MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>            ; 5000
003030' 000000  004040
003031' 500150  010020                    MWORD   <JMAP,J=5001,SELM,MGC=20>                       ; 5001
003032' 000000  002040
003033' 777777  777777                    -1
```

```
                                        ;#***************************************************************************
                                        ;* Test 11 - Fmtr Cleared by 'Port Clear'
                                        ;
                                        ; Description:  A 'Port Clear' should clear the contents  of  the
                                        ;               Mover/Formatter.  This test verifies this.
                                        ;
                                        ; Procedure:    KL> Port Clear
                                        ;               UC> Read FMTR, place data in 2901's
                                        ;               UC> Move data from 2901's to EBUF
                                        ;               KL> Read EBUF and verify the data is zero
                                        ;
                                        ; Failure:      ---
                                        ;#***************************************************************************

                                        ; Test data

003034' 254 00 0 00 003045'             TSTC11: JRST    TG11                    ; go start test
003035' 100604  000011                          CBUS!NDMP!NDCB!ZCBUS!11         ; test mask
003036' 003074' 013227'                         T11M,,[ASCIZ ^Fmtr Cleared by 'Port Clear'^]
003037' 013235' 013237'                         [EXP C20,MLAST!C4],,[EXP C12,C8,C18,C23,C24,C3,MLAST!C2]
003040' 000000  001044'                         TSTC3                           ; failure test table
003041' 000000  001152'                         TSTC4                           ;   ...
003042' 000000  001275'                         TSTC5
003043' 000000  001612'                         TSTC6
003044' 777777  777777                          -1

                                        ; Start test

003045' 201 00 0 00 000000'             TG11:   MOVEI   Z9                      ; get address of module start
003046' 260 17 0 00 002527*                     GO      TRACE                   ; handle trace output
003047' 201 01 0 00 003074'                     MOVEI   1,T11M                  ; set up microcode address
003050' 260 17 0 00 002531*                     GO      TLOAD                   ; load/verify it
003051' 263 17 0 00 000000                      RTN                             ; failed - exit test

                                        ; Initialization

003052' 400 15 0 00 000000              TL11:   SETZ    ERFLG,                  ; clear error flag
003053' 260 17 0 00 002617*                     GO      IPACLR                  ; clear port
003054' 402 00 0 00 002535*                     SETZM   TSTSUB                  ; initialize subtest number
003055' 201 06 0 00 003067'                     MOVEI   6,TS11                  ; get sstep table address

                                        ; Loop on single srep table entries

003056' 260 17 0 00 000657*             TA11:   GO      BEXEC                   ; execute table entry
003057' 254 00 0 00 003066'                     JRST    TX11                    ; end of sstep table
003060' 254 00 0 00 003056'                     JRST    TA11                    ; keep looping after call
003061' 474 15 0 00 000000                      SETO    ERFLG,                  ; error - EBUF data incorrect
```

```

                                          ; Handle error printouts and scope looping

003062' 027 00 0 00 003072'                     SCOPER  MA11                    ; print error message
003063' 254 00 0 00 003052'                     JRST    TL11                    ; loop on error
003064' 254 00 0 00 003066'                     JRST    TX11                    ; altmode exit
003065' 322 15 0 00 003056'                     JUMPE   ERFLG,TA11              ; do next sstep table entry

                                          ; End of test

003066' 263 17 0 00 000000                TX11: RTN                             ; return

                                          ; Sstep table, with EBUF data, as:  (CMD,#Steps,Start addr,End Addr,EBUF data)

003067' 100200  000001                    TS11: ATABLE  (SSSTRT,2,0,1,0)
003070' 000000  000000
003071' 000000  000000                          ATABLE  (SSLAST)

                                          ; Error messages

003072' 140000  013246'                   MA11: MSG!TXNOT![ASCIZ /Fmtr not cleared by a 'Port Clear'/]
003073' 000000000000#                           LAST!CALL!TXALL!BBPNT           ; print sequencer data

                                          ; Microcode:

003074' 000000  010200                    T11M: MWORD   <ADDR=0,JMAP,J=1,SDO,OR,D=2,SELM,MGC=200>
003075' 732000  002040
003076' 000100  012004                          MWORD   <JMAP,J=1,SOA,OR,D=1,OENA,SELE,MGC=4>   ; 1
003077' 431000  005040
003100' 777777  777777                          -1
```

```
                                 ;#*********************************************************************
                                 ;* Test 12 - Fmtr Basic Up Shifting Test
                                 ;
                                 ; Description:  Verify that the  Mover/Formatter  is  capable  of
                                 ;               being shifted up.  The contents of the  formatter
                                 ;               is initially zero, and the result of any shifting
                                 ;               should be zero.
                                 ;
                                 ; Procedure:    KL> Port Clear
                                 ;               UC> Shift formatter 4 bits up
                                 ;               UC> Read formatter data
                                 ;               UC> If zero, continue.
                                 ;                   If non-zero, halt with a CRAM PE and leave
                                 ;                     the non-zero data read in the EBUF.
                                 ;               KL> Let it run at full speed for 1/2 second then
                                 ;                   halt and verify that no CRAM PE occurred and
                                 ;                   that the microcode did not stop at the error
                                 ;                   location.
                                 ;
                                 ; Failure:      ---
                                 ;#*********************************************************************

                                 ; Test data

003101' 254 00 0 00 003112'      TSTC12: JRST    TG12                    ; go start test
003102' 100604  000012                   CBUS!NDMP!NDCB!ZCBUS!12         ; test mask
003103' 003175' 013255'                  T12M,,[ASCIZ ^Fmtr Basic Up Shifting Test^]
003104' 013263' 012716'                  [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
003105' 000000  003210'                  TSTC13                          ; failure test table
003106' 000000  003510'                  TSTC14                          ;   ...
003107' 000000  006754'                  TSTC25
003110' 000000  007062'                  TSTC26
003111' 777777  777777                   -1

                                 ; Start test

003112' 201 00 0 00 000000'      TG12:   MOVEI   Z9                      ; get address of module start
003113' 260 17 0 00 003046*              GO      TRACE                   ; handle trace output
003114' 201 01 0 00 003175'              MOVEI   1,T12M                  ; set up microcode address
003115' 260 17 0 00 003050*              GO      TLOAD                   ; load/verify it
003116' 263 17 0 00 000000               RTN                             ; failed - exit test
```

```
                                        ; 1st segment of test (Segment A) - Start up port

003117' 400 15 0 00 000000      TA12:   SETZ    ERFLG,                  ; clear error flag
003120' 201 00 0 00 000001              MOVEI   1                       ; initialize subtest number
003121' 202 00 0 00 003054*             MOVEM   TSTSUB                  ;   to 1
003122' 260 17 0 00 003053*             GO      IPACLR                  ; issue a port clear
003123' 402 00 0 00 000000*             SETZM   SNEXT                   ; set up start location
003124' 201 00 0 00 000010              MOVEI   10                      ; get a 10
003125' 202 00 0 00 000000*             MOVEM   SDATA                   ; set up start CSR data
003126' 260 17 0 00 000000*             GO      IPASRT                  ; start up the port
003127' 474 15 0 00 000000              SETO    ERFLG,                  ; failed - error reading/writing CSR
003130' 474 15 0 00 000000              SETO    ERFLG,                  ; failed - port already running
003131' 474 15 0 00 000000              SETO    ERFLG,                  ; failed - error bits set in CSR
003132' 027 00 0 00 003155'             SCOPER  MA12                    ; print error message
003133' 254 00 0 00 003117'             JRST    TA12                    ; loop on error
003134' 254 00 0 00 003154'             JRST    TX12                    ; altmode exit
003135' 326 15 0 00 003154'             JUMPN   ERFLG,TX12              ; error already - exit test

                                        ; 2nd segment of test (Segment B) - Check if CRAM PE sets after 1/2 second

003136' 350 00 0 00 003121*             AOS     TSTSUB                  ; increment subtest number
003137' 201 00 0 00 000764              MOVEI   ^D500                   ; wait 1/2 second
003140' 260 17 0 00 002567*             GO      ODELAY                  ; do the wait
003141' 260 17 0 00 000000*             GO      RDCSR                   ; read CSR
003142' 474 15 0 00 000000              SETO    ERFLG,                  ; failed - couldn't read CSR
003143' 603 01 0 00 004000              TLNE    1,(CRAMPE)              ; got a CRAM PE?
003144' 474 15 0 00 000000              SETO    ERFLG,                  ; yes - set error flag
003145' 201 01 0 00 204000              MOVEI   1,EBUSPE!TSTEBF         ; get bits to clear error condition
003146' 260 17 0 00 002565*             GO      LDCSR                   ;   and set up to read the EBUF
003147' 260 17 0 00 001771*             GO      RDEBUF                  ; read EBUF data
003150' 202 01 0 00 014013'             MOVEM   1,SAVDAT#               ; save data
003151' 027 00 0 00 003167'             SCOPER  MB12                    ; print error message
003152' 254 00 0 00 003117'             JRST    TA12                    ; loop on error
003153' 254 00 0 00 003154'             JRST    TX12                    ; altmode exit

003154' 263 17 0 00 000000      TX12:   RTN                             ; exit test
```

```

                                           ; Error messages

003155' 140000  013266'                    MA12:    MSG!TXNOT![ASCIZ /Couldn't successfully start the port/]
003156' 270000  003157'                             LAST!CALL!TXALL!MA12P1

003157' 260 17 0 00 003141*                MA12P1:  GO       RDCSR                    ; get CSR contents
003160' 255 00 0 00 000000                          JFCL                              ; error
003161' 202 01 0 00 014010'                         MOVEM    1,SAVCSR#                ; save it
003162' 200 01 0 00 014010'                         MOVE     1,SAVCSR                 ; get CSR data
003163' 201 02 0 00 000023                          MOVEI    2,^D19                   ; character count so far
003164' 201 03 0 00 000023                          MOVEI    3,^D19                   ; continuation line column number
003165' 260 17 0 00 000000*                         GO       CSRPNT                   ; go print in English
003166' 263 17 0 00 000000                          RTN                               ; return

003167' 140000  013276'                    MB12:    MSG!TXNOT![ASCIZ /Up shifting of zero data resulted in non-zero data/]
003170' 270000  003171'                             LAST!CALL!TXALL!MA12PN

003171' 037 00 0 00 013311'                MA12PN:  TMSGC    <EBUF Data (Formatter contents):  >
003172' 200 00 0 00 014013'                         MOVE     SAVDAT
003173' 037 13 0 00 000000                          PNTHW
003174' 263 17 0 00 000000                          RTN

                                           ; Microcode:

003175' 000000  010100                     T12M:    MWORD    <ADDR=0,JMAP,J=1,SOA,D=1,SELF,MGC=100>  ; 0
003176' 401000  003040
003177' 000100  000200                              MWORD    <CONT,SDO,OR,D=2,SELM,MGC=200>          ; 1
003200' 732000  002340
003201' 000200  000000                              MWORD    <CJP,J=0,SOA,OR,D=1,CENA,CCFZ>          ; 2
003202' 431400  020060
003203' 000300  002004                              MWORD    <CONT,SOA,OR,D=1,CENA,SELE,MGC=4>       ; 3
003204' 431000  005340
003205' 000400  040000                              MWORD    <JMAP,J=4,BAD>                          ; 4
003206' 000000  000041
003207' 777777  777777                              -1
```

```
                                        ;#*********************************************************************
                                        ;* Test 13 - Fmtr Up Shift 4 Test - BUS Ctl=Cbus
                                        ;
                                        ; Description:  Verify that the formatter can shift various  data
                                        ;               patterns up 4 bits at a time.
                                        ;
                                        ; Procedure:    KL> Port Clear
                                        ;               KL> Set up data transfer for data patterns
                                        ;
                                        ;               UC> Read a data pattern over the CBUS
                                        ;               UC> Shift up 4 bits
                                        ;               UC> Read formatter contents and write into
                                        ;                   EBUF
                                        ;               KL> Read EBUF and verify the data
                                        ;               KL> Start up the ucode to shift again
                                        ;
                                        ;               Repeat the above shifting then verifying EBUF
                                        ;                 contents 10 times, until the data becomes
                                        ;                 all zero.
                                        ;
                                        ;               Do for data patterns:     0's
                                        ;                                         1's
                                        ;                                         525252,,525252
                                        ;                                         000000,,007777
                                        ;                                         000000,,007070
                                        ;                                         000000,,000707
                                        ;
                                        ; Failure:      ---
                                        ;#*********************************************************************

                                        ; Test data

003210' 254 00 0 00 003220'             TSTC13: JRST    TG13                            ; go start test
003211' 100604  000013                          CBUS!NDMP!NDCB!ZCBUS!13                 ; test mask
003212' 003427' 013321'                         T13M,,[ASCIZ ^Fmtr Up Shift 4 Test - BUS Ctl=Cbus^]
003213' 013263' 012716'                         [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
003214' 000000  003510'                         TSTC14                                  ; failure test table
003215' 000000  006754'                         TSTC25                                  ;   ...
003216' 000000  007062'                         TSTC26
003217' 777777  777777                          -1

                                        ; Start test

003220' 201 00 0 00 000000'             TG13:   MOVEI   Z9                              ; get address of module start
003221' 260 17 0 00 003113*                     GO      TRACE                           ; handle trace output
003222' 201 01 0 00 003427'                     MOVEI   1,T13M                          ; set up microcode address
003223' 260 17 0 00 003115*                     GO      TLOAD                           ; load/verify it
003224' 263 17 0 00 000000                      RTN                                     ; failed - exit test
```

```
                                        ; Initialization

        003225' 400 15 0 00 000000      TL13:   SETZ    ERFLG,                  ; clear error flag
        003226' 260 17 0 00 003122*             GO      IPACLR                  ; clear port
        003227' 402 00 0 00 003136*             SETZM   TSTSUB                  ; initialize subtest number
                                                MOVSI   [000000,,000000         ; get address of buffer
                                                         777777,,777777         ;   contents
                                                         525252,,525252
                                                         000000,,007777
                                                         000000,,007070
        003230' 205 00 0 00 013330'                      000000,,000707]
        003231' 541 00 0 00 002626*             HRRI    BUFF                    ; build the BLT pointer
        003232' 251 00 0 00 000000#             BLT     BUFF+5                  ; insert data into buffer

                                        ; First ensure the data transfer is reset

        003233' 201 01 0 00 005000              MOVEI   1,5000                  ; get start address
        003234' 242 01 0 00 000001              LSH     1,1                     ; position correctly
        003235' 260 17 0 00 002563*             GO      LDRAR                   ; load the register
        003236' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get 'MPRUN' bit
        003237' 260 17 0 00 003146*             GO      LDCSR                   ; start up the port
        003240' 400 01 0 00 000000              SETZ    1,                      ; stop the port
        003241' 260 17 0 00 003237*             GO      LDCSR

                                        ; Now generate CCW list

        003242' 200 01 0 00 013210'             MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
        003243' 260 17 0 00 002553*             GO      $INIT                   ; initialize software
        003244' 551 01 0 00 003231*             HRRZI   1,BUFF                  ; buffer address
        003245' 201 02 0 00 000006              MOVEI   2,6                     ; word count
        003246' 400 03 0 00 000000              SETZ    3,                      ; clear zero fill specifier
        003247' 260 17 0 00 002557*             GO      GENCCW                  ; generate a CCW list
        003250' 201 06 0 00 003273'             MOVEI   6,TS13                  ; get test execute table address

                                        ; Start up the data transfer

        003251' 201 01 0 00 001000              MOVEI   1,1000                  ; get start address
        003252' 242 01 0 00 000001              LSH     1,1                     ; position correctly
        003253' 260 17 0 00 003235*             GO      LDRAR                   ; load the register
        003254' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
        003255' 260 17 0 00 003241*             GO      LDCSR                   ; start up the port
        003256' 201 00 0 00 000001              MOVEI   1                       ; set up delay of 1 ms
        003257' 260 17 0 00 003140*             GO      ODELAY                  ; wait a bit
        003260' 400 01 0 00 000000              SETZ    1,                      ; clear data
        003261' 260 17 0 00 003255*             GO      LDCSR                   ; stop the port
```

```

                                        ; Loop on test execute table entries

003262' 260 17 0 00 000000*     TA13:   GO      TEXEC                   ; execute table entry
003263' 254 00 0 00 003272'             JRST    TX13                    ; end of table
003264' 254 00 0 00 003262'             JRST    TA13                    ; keep looping after call
003265' 474 15 0 00 000000              SETO    ERFLG,                  ; error - EBUF data incorrect

                                ; Handle error printouts and scope looping

003266' 027 00 0 00 003414'             SCOPER  MA13                    ; print error message
003267' 254 00 0 00 003225'             JRST    TL13                    ; loop on error
003270' 254 00 0 00 003272'             JRST    TX13                    ; altmode exit
003271' 322 15 0 00 003262'             JUMPE   ERFLG,TA13              ; do next sstep table entry

                                ; End of test

003272' 263 17 0 00 000000      TX13:   RTN                             ; return

                                ; Test Execute Table, as:  (CMD,parameters)

003273' 140000  003347'         TS13:   TTABLE  (TCALL,TS13IN)          ; initialize stuff
003274' 200000  003352'         TS13L:  TTABLE  (TCALLC,TS13NE)         ; set up next data pattern

                                ; Shifted 4 bits

003275' 140000  003377'         TS13A:  TTABLE  (TCALL,TS13NN)          ; increment shift count
003276' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
003277' 240001  003402'                 TTABLE  (TCHECK,1,TS13RS)       ; check if completed
003300' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                ; Shifted 8 bits

003301' 140000  003377'                 TTABLE  (TCALL,TS13NN)          ; increment shift count
003302' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
003303' 240001  003403'                 TTABLE  (TCHECK,1,TS13RS+1)     ; check if completed
003304' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                ; Shifted 12 bits

003305' 140000  003377'                 TTABLE  (TCALL,TS13NN)          ; increment shift count
003306' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
003307' 240001  003404'                 TTABLE  (TCHECK,1,TS13RS+2)     ; check if completed
003310' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                ; Shifted 16 bits

003311' 140000  003377'                 TTABLE  (TCALL,TS13NN)          ; increment shift count
003312' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
003313' 240001  003405'                 TTABLE  (TCHECK,1,TS13RS+3)     ; check if completed
003314' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

```

```

                                        ; Shifted 20 bits

        003315' 140000  003377'                 TTABLE  (TCALL,TS13NN)          ; increment shift count
        003316' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
        003317' 240001  003406'                 TTABLE  (TCHECK,1,TS13RS+4)     ; check if completed
        003320' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 24 bits

        003321' 140000  003377'                 TTABLE  (TCALL,TS13NN)          ; increment shift count
        003322' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
        003323' 240001  003407'                 TTABLE  (TCHECK,1,TS13RS+5)     ; check if completed
        003324' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 28 bits

        003325' 140000  003377'                 TTABLE  (TCALL,TS13NN)          ; increment shift count
        003326' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
        003327' 240001  003410'                 TTABLE  (TCHECK,1,TS13RS+6)     ; check if completed
        003330' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 32 bits

        003331' 140000  003377'                 TTABLE  (TCALL,TS13NN)          ; increment shift count
        003332' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
        003333' 240001  003411'                 TTABLE  (TCHECK,1,TS13RS+7)     ; check if completed
        003334' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 36 bits

        003335' 140000  003377'                 TTABLE  (TCALL,TS13NN)          ; increment shift count
        003336' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
        003337' 240001  003412'                 TTABLE  (TCHECK,1,TS13RS+10)    ; check if completed
        003340' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 40 bits

        003341' 140000  003377'                 TTABLE  (TCALL,TS13NN)          ; increment shift count
        003342' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
        003343' 240001  003413'                 TTABLE  (TCHECK,1,TS13RS+11)    ; check if completed
        003344' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet
        003345' 300000  003274'                 TTABLE  (TJRST,TS13L)           ; loop till done
        003346' 000000  000000                  TTABLE  (TLAST)

                                        ; Initialize stuff

        003347' 201 00 0 00 000000#     TS13IN: MOVEI   BUFF-1                  ; initialize buffer pointer
        003350' 202 00 0 00 014015'             MOVEM   T13BUF#                 ;  ...
        003351' 263 17 0 00 000000              RTN                             ; return
```

```


                                ; Set up next data pattern

003352' 350 01 0 00 014015'     TS13NE: AOS     1,T13BUF                ; point to next data pattern
003353' 307 01 0 00 000000#             CAIG    1,BUFF+5                ; done yet?
003354' 350 00 0 17 000000              AOS     (P)                     ; no - set up proper return
003355' 200 00 0 01 000000              MOVE    (1)                     ; get data pattern
003356' 201 02 0 00 003402'             MOVEI   2,TS13RS                ; get initial address of EBUF data
003357' 242 00 0 00 000004      TS13NO: LSH     4                       ; shift 4 bits
003360' 202 00 0 02 000000              MOVEM   (2)                     ; save it
003361' 350 00 0 00 000002              AOS     2                       ; point to next address
003362' 307 02 0 00 003413'             CAIG    2,TS13RS+11             ; done yet?
003363' 254 00 0 00 003357'             JRST    TS13NO                  ; no - loop till done
003364' 402 00 0 00 014040'             SETZM   TS13NM#                 ; initialize number of shifts

                                ; Read the next word

003365' 400 01 0 00 000000              SETZ    1,                      ; clear data
003366' 260 17 0 00 003261*             GO      LDCSR                   ; stop the port
003367' 201 01 0 00 000200              MOVEI   1,200                   ; get start address
003370' 242 01 0 00 000001              LSH     1,1                     ; position correctly
003371' 260 17 0 00 003253*             GO      LDRAR                   ; load the register
003372' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
003373' 260 17 0 00 003366*             GO      LDCSR                   ; start up the port
003374' 400 01 0 00 000000              SETZ    1,                      ; clear data
003375' 260 17 0 00 003373*             GO      LDCSR                   ; stop the port
003376' 263 17 0 00 000000              RTN                             ; return

                                ; Increment shift count

003377' 201 00 0 00 000004      TS13NN: MOVEI   4                       ; get a 4
003400' 272 00 0 00 014040'             ADDM    TS13NM                  ; add it
003401' 263 17 0 00 000000              RTN                             ; return

003402'                         TS13RS: BLOCK   12                      ; correct EBUF data

                                ; Error messages

003414' 240000  003416'         MA13:   CALL!TXNOT!MA13PN               ; print data pattern ...
003415' 000000000000#                   LAST!CALL!TXALL!TTPNT           ; print test data

003416' 037 00 0 00 013336'     MA13PN: TMSGC   <Data >                 ; get data and
003417' 200 01 0 00 014015'             MOVE    1,T13BUF                ;   print it
003420' 200 00 0 01 000000              MOVE    (1)
003421' 037 13 0 00 000000              PNTHW
003422' 037 00 0 00 013340'             TMSG    < shifted up >
003423' 200 00 0 00 014040'             MOVE    TS13NM                  ; get shift count
003424' 037 15 0 00 000000              PNTDEC
003425' 037 00 0 00 013343'             TMSG    <. bits, result in EBUF>
003426' 263 17 0 00 000000              RTN
```

```
                                    ; Microcode - Shift 4 bits

003427' 010001  010000              T13M:   MWORD   <ADDR=100,JMAP,J=101,D=1>                 ; 100
003430' 001000  000040
003431' 010100  000001                      MWORD   <CONT,D=1,SELC,MGC=1>                     ; 101
003432' 001000  004340
003433' 010200  000200                      MWORD   <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>        ; 102
003434' 732002  002340
003435' 010351  000000                      MWORD   <JMAP,J=5100,D=1>                         ; 103
003436' 001000  000040

                                    ;       Routine to read a word from CBUS into the formatter.

003437' 020002  010000                      MWORD   <ADDR=200,JMAP,J=201>                     ; 200
003440' 000000  000040
003441' 020102  030000                      MWORD   <CJP,J=203,D=1,CENA,CCCA>                 ; 201
003442' 001400  000060
003443' 020202  010000                      MWORD   <JMAP,J=201,D=1>                          ; 202
003444' 001000  000040
003445' 020300  000004                      MWORD   <CONT,D=1,SELC,MGC=4>                     ; 203
003446' 001000  004340
003447' 020400  000200                      MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>        ; 204
003450' 732001  402340
003451' 020500  000000                      MWORD   <JZ>                                      ; 205
003452' 000000  000000

                                    ;       Initialization

003453' 100010  010020                      MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>       ; 1000
003454' 000000  002040
003455' 100100  000200                      MWORD   <JZ,D=1,SELC,MGC=200>                     ; 1001
003456' 001000  004000
003457' 000000  000000                      MWORD   <ADDR=0,JZ>                               ; 0
003460' 000000  000000

                                    ;       Initialization step

003461' 500050  010100                      MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>      ; 5000
003462' 000000  004040
003463' 500100  000020                      MWORD   <JZ,SELM,MGC=20>                          ; 5001
003464' 000000  002000

                                    ; Subroutine - Passed - Write 1 to bit 12 of CSR register

003465' 510000  000000                      MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>          ; 5100
003466' 742001  000340
003467' 510100  000000                      MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>               ; 5101
003470' 302001  000740
003471' 510200  260000                      MWORD   <LDCT,J=26>                               ; 5102
003472' 000000  000300
003473' 510351  030000                      MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>        ; 5103
003474' 102021  000220
003475' 510400  002004                      MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>     ; 5104
003476' 431020  005340
```

```
 2907  003477' 510551  070040            MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>              ; 5105
 2908  003500' 001400  015060
 2909  003501' 510651  050000            MWORD   <JMAP,J=5105,D=1>                                    ; 5106
 2910  003502' 001000  000040
 2911  003503' 510700  000200            MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>                  ; 5107
 2912  003504' 431010  005340
 2913  003505' 511051  102004            MWORD   <JMAP,J=5110,SOA,OR,A=4,D=1,OENA,SELE,MGC=4>        ; 5110
 2914  003506' 431040  005040
 2915  003507' 777777  777777            -1
```

DFPTA9
DFPTA9

```
347
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
```

```
                                        ;#*****************************************************************
                                        ;* Test 14 - Fmtr Up Shift 4 Test - BUS Ctl=Fmtr
                                        ;
                                        ; Description:  Verify that the formatter can shift various  data
                                        ;               patterns up 4 bits at a time.
                                        ;
                                        ; Procedure:    KL> Port Clear
                                        ;               KL> Set up data transfer for data patterns
                                        ;
                                        ;               UC> Read a data pattern over the CBUS
                                        ;               UC> Shift up 4 bits
                                        ;               UC> Read formatter contents and write into
                                        ;                   EBUF
                                        ;               KL> Read EBUF and verify the data
                                        ;               KL> Start up the ucode to shift again
                                        ;
                                        ;               Repeat the above shifting then verifying EBUF
                                        ;                 contents 10 times, until the data becomes
                                        ;                 all zero.
                                        ;
                                        ;               Do for data patterns:   0's
                                        ;                                       1's
                                        ;                                       525252,,525252
                                        ;                                       000000,,007777
                                        ;                                       000000,,007070
                                        ;                                       000000,,000707
                                        ;
                                        ; Failure:      ---
                                        ;#*****************************************************************

                                        ; Test data

003510' 254 00 0 00 003521'             TSTC14: JRST    TG14                    ; go start test
003511' 100604  000014                          CBUS!NDMP!NDCB!ZCBUS!14         ; test mask
003512' 003726' 013350'                         T14M,,[ASCIZ ^Fmtr Up Shift 4 Test - BUS Ctl=Fmtr^]
003513' 013263' 012716'                         [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
003514' 000000  004007'                         TSTC15                          ; failure test table
003515' 000000  004322'                         TSTC16                          ;   ...
003516' 000000  006754'                         TSTC25
003517' 000000  007062'                         TSTC26
003520' 777777  777777                          -1

                                        ; Start test

003521' 201 00 0 00 000000'             TG14:   MOVEI   Z9                      ; get address of module start
003522' 260 17 0 00 003221*                     GO      TRACE                   ; handle trace output
003523' 201 01 0 00 003726'                     MOVEI   1,T14M                  ; set up microcode address
003524' 260 17 0 00 003223*                     GO      TLOAD                   ; load/verify it
003525' 263 17 0 00 000000                      RTN                             ; failed - exit test
```

```

                                ; Initialization

003526' 400 15 0 00 000000      TL14:   SETZ    ERFLG,                  ; clear error flag
003527' 260 17 0 00 003226*             GO      IPACLR                  ; clear port
003530' 402 00 0 00 003227*             SETZM   TSTSUB                  ; initialize subtest number
                                        MOVSI   [000000,,000000         ; get address of buffer
                                                 777777,,777777         ;   contents
                                                 525252,,525252
                                                 000000,,007777
                                                 000000,,007070
003531' 205 00 0 00 013330'                      000000,,000707]
003532' 541 00 0 00 003244*             HRRI    BUFF                    ; build the BLT pointer
003533' 251 00 0 00 000000#             BLT     BUFF+5                  ; insert data into buffer

                                ; First ensure the data transfer is reset

003534' 201 01 0 00 005000              MOVEI   1,5000                  ; get start address
003535' 242 01 0 00 000001              LSH     1,1                     ; position correctly
003536' 260 17 0 00 003371*             GO      LDRAR                   ; load the register
003537' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
003540' 260 17 0 00 003375*             GO      LDCSR                   ; start up the port
003541' 400 01 0 00 000000              SETZ    1,                      ; clear data
003542' 260 17 0 00 003540*             GO      LDCSR                   ; stop the port

                                ; Now generate CCW list

003543' 200 01 0 00 013210'             MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
003544' 260 17 0 00 003243*             GO      CHINIT                  ; initialize software
003545' 551 01 0 00 003532*             HRRZI   1,BUFF                  ; buffer address
003546' 201 02 0 00 000006              MOVEI   2,6                     ; word count
003547' 400 03 0 00 000000              SETZ    3,                      ; clear zero fill specifier
003550' 260 17 0 00 003247*             GO      GENCCW                  ; generate a CCW list
003551' 201 06 0 00 003574'             MOVEI   6,TS14                  ; get test execute table address

                                ; Start up the data transfer

003552' 201 01 0 00 001000              MOVEI   1,1000                  ; get start address
003553' 242 01 0 00 000001              LSH     1,1                     ; position correctly
003554' 260 17 0 00 003536*             GO      LDRAR                   ; load the register
003555' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
003556' 260 17 0 00 003542*             GO      LDCSR                   ; start up the port
003557' 201 00 0 00 000001              MOVEI   1                       ; set up delay of 1 ms
003560' 260 17 0 00 003257*             GO      ODELAY                  ; wait a bit
003561' 400 01 0 00 000000              SETZ    1,                      ; clear data
003562' 260 17 0 00 003556*             GO      LDCSR                   ; stop the port
```

```

                                        ; Loop on test execute table entries

003563' 260 17 0 00 003262*     TA14:   GO      TEXEC                   ; execute table entry
003564' 254 00 0 00 003573'             JRST    TX14                    ; end of table
003565' 254 00 0 00 003563'             JRST    TA14                    ; keep looping after call
003566' 474 15 0 00 000000              SETO    ERFLG,                  ; error - EBUF data incorrect

                                        ; Handle error printouts and scope looping

003567' 027 00 0 00 003713'             SCOPER  MA14                    ; print error message
003570' 254 00 0 00 003526'             JRST    TL14                    ; loop on error
003571' 254 00 0 00 003573'             JRST    TX14                    ; altmode exit
003572' 322 15 0 00 003563'             JUMPE   ERFLG,TA14              ; do next sstep table entry

                                        ; End of test

003573' 263 17 0 00 000000      TX14:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

003574' 140000  003650'         TS14:   TTABLE  (TCALL,TS14IN)          ; initialize stuff
003575' 200000  003653'         TS14L:  TTABLE  (TCALLC,TS14NE)         ; set up next data pattern

                                        ; Shifted 4 bits

003576' 140000  003676'         TS14A:  TTABLE  (TCALL,TS14NN)          ; increment shift count
003577' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
003600' 240001  003701'                 TTABLE  (TCHECK,1,TS14RS)       ; check if completed
003601' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 8 bits

003602' 140000  003676'                 TTABLE  (TCALL,TS14NN)          ; increment shift count
003603' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
003604' 240001  003702'                 TTABLE  (TCHECK,1,TS14RS+1)     ; check if completed
003605' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 12 bits

003606' 140000  003676'                 TTABLE  (TCALL,TS14NN)          ; increment shift count
003607' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
003610' 240001  003703'                 TTABLE  (TCHECK,1,TS14RS+2)     ; check if completed
003611' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 16 bits

003612' 140000  003676'                 TTABLE  (TCALL,TS14NN)          ; increment shift count
003613' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
003614' 240001  003704'                 TTABLE  (TCHECK,1,TS14RS+3)     ; check if completed
003615' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

```

```

                                ; Shifted 20 bits

003616' 140000  003676'                 TTABLE  (TCALL,TS14NN)          ; increment shift count
003617' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
003620' 240001  003705'                 TTABLE  (TCHECK,1,TS14RS+4)     ; check if completed
003621' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                ; Shifted 24 bits

003622' 140000  003676'                 TTABLE  (TCALL,TS14NN)          ; increment shift count
003623' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
003624' 240001  003706'                 TTABLE  (TCHECK,1,TS14RS+5)     ; check if completed
003625' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                ; Shifted 28 bits

003626' 140000  003676'                 TTABLE  (TCALL,TS14NN)          ; increment shift count
003627' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
003630' 240001  003707'                 TTABLE  (TCHECK,1,TS14RS+6)     ; check if completed
003631' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                ; Shifted 32 bits

003632' 140000  003676'                 TTABLE  (TCALL,TS14NN)          ; increment shift count
003633' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
003634' 240001  003710'                 TTABLE  (TCHECK,1,TS14RS+7)     ; check if completed
003635' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                ; Shifted 36 bits

003636' 140000  003676'                 TTABLE  (TCALL,TS14NN)          ; increment shift count
003637' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
003640' 240001  003711'                 TTABLE  (TCHECK,1,TS14RS+10)    ; check if completed
003641' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                ; Shifted 40 bits

003642' 140000  003676'                 TTABLE  (TCALL,TS14NN)          ; increment shift count
003643' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
003644' 240001  003712'                 TTABLE  (TCHECK,1,TS14RS+11)    ; check if completed
003645' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet
003646' 300000  003575'                 TTABLE  (TJRST,TS14L)           ; loop till done
003647' 000000  000000                  TTABLE  (TLAST)

                                ; Initialize stuff

003650' 201 00 0 00 000000#     TS14IN: MOVEI   BUFF-1                  ; initialize buffer pointer
003651' 202 00 0 00 014016'             MOVEM   T14BUF#                 ;  ...
003652' 263 17 0 00 000000              RTN                             ; return
```

```

                                          ; Set up next data pattern

003653' 350 01 0 00 014016'               TS14NE: AOS     1,T14BUF                  ; point to next data pattern
003654' 307 01 0 00 000000#                       CAIG    1,BUFF+5                  ; done yet?
003655' 350 00 0 17 000000                        AOS     (P)                       ; no - set up proper return
003656' 200 00 0 01 000000                        MOVE    (1)                       ; get data pattern
003657' 201 02 0 00 003701'                       MOVEI   2,TS14RS                  ; get initial address of EBUF data
003660' 242 00 0 00 000004                TS14NO: LSH     4                         ; shift 4 bits
003661' 202 00 0 02 000000                        MOVEM   (2)                       ; save it
003662' 350 00 0 00 000002                        AOS     2                         ; point to next address
003663' 307 02 0 00 003712'                       CAIG    2,TS14RS+11               ; done yet?
003664' 254 00 0 00 003660'                       JRST    TS14NO                    ; no - loop till done
003665' 402 00 0 00 014041'                       SETZM   TS14NM#                   ; initialize number of shifts

                                          ; Read the next word

003666' 201 01 0 00 000200                        MOVEI   1,200                     ; get start address
003667' 242 01 0 00 000001                        LSH     1,1                       ; position correctly
003670' 260 17 0 00 003554*                       GO      LDRAR                     ; load the register
003671' 201 01 0 00 000010                        MOVEI   1,MPRUN                   ; get MPRUN bit
003672' 260 17 0 00 003562*                       GO      LDCSR                     ; start up the port
003673' 400 01 0 00 000000                        SETZ    1,                        ; clear data
003674' 260 17 0 00 003672*                       GO      LDCSR                     ; stop the port
003675' 263 17 0 00 000000                        RTN                               ; return

                                          ; Increment shift count

003676' 201 00 0 00 000004                TS14NN: MOVEI   4                         ; get a 4
003677' 272 00 0 00 014041'                       ADDM    TS14NM                    ; add it
003700' 263 17 0 00 000000                        RTN                               ; return

003701'                                   TS14RS: BLOCK   12                        ; correct EBUF data

                                          ; Error messages

003713' 240000  003715'                   MA14:   CALL!TXNOT!MA14PN                 ; print data pattern ...
003714' 000000000000#                             LAST!CALL!TXALL!TTPNT             ; print test data

003715' 037 00 0 00 013336'               MA14PN: TMSGC   <Data >                   ; get data and
003716' 200 01 0 00 014016'                       MOVE    1,T14BUF                  ;   print it
003717' 200 00 0 01 000000                        MOVE    (1)
003720' 037 13 0 00 000000                        PNTHW
003721' 037 00 0 00 013340'                       TMSG    < shifted up >
003722' 200 00 0 00 014041'                       MOVE    TS14NM                    ; get shift count
003723' 037 15 0 00 000000                        PNTDEC
003724' 037 00 0 00 013343'                       TMSG    <. bits, result in EBUF>
003725' 263 17 0 00 000000                        RTN
```

```

                                        ; Microcode - Shift 4 bits

003726' 010001  010000                  T14M:   MWORD   <ADDR=100,JMAP,J=101,D=1>                       ; 100
003727' 001000  000040
003730' 010100  000100                          MWORD   <CONT,D=1,SELF,MGC=100>                         ; 101
003731' 001000  003340
003732' 010200  000200                          MWORD   <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>              ; 102
003733' 732002  002340
003734' 010351  000000                          MWORD   <JMAP,J=5100,D=1>                               ; 103
003735' 001000  000040

                                        ;       Routine to read a word from CBUS into the formatter.

003736' 020002  010000                          MWORD   <ADDR=200,JMAP,J=201>                           ; 200
003737' 000000  000040
003740' 020102  030000                          MWORD   <CJP,J=203,D=1,CENA,CCCA>                       ; 201
003741' 001400  000060
003742' 020202  010000                          MWORD   <JMAP,J=201,D=1>                                ; 202
003743' 001000  000040
003744' 020300  000004                          MWORD   <CONT,D=1,SELC,MGC=4>                           ; 203
003745' 001000  004340
003746' 020400  000200                          MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>              ; 204
003747' 732001  402340
003750' 020500  000000                          MWORD   <JZ>                                            ; 205
003751' 000000  000000

                                        ;       Initialization

003752' 100010  010020                          MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>             ; 1000
003753' 000000  002040
003754' 100100  000200                          MWORD   <JZ,D=1,SELC,MGC=200>                           ; 1001
003755' 001000  004000
003756' 000000  000000                          MWORD   <ADDR=0,JZ>                                     ; 0
003757' 000000  000000

                                        ;       Initialization step

003760' 500050  010100                          MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>            ; 5000
003761' 000000  004040
003762' 500100  000020                          MWORD   <JZ,SELM,MGC=20>                                ; 5001
003763' 000000  002000

                                        ; Subroutine - Passed - Write 1 to bit 12 of CSR register

003764' 510000  000000                          MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>                ; 5100
003765' 742001  000340
003766' 510100  000000                          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                     ; 5101
003767' 302001  000740
003770' 510200  260000                          MWORD   <LDCT,J=26>                                     ; 5102
003771' 000000  000300
003772' 510351  030000                          MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>              ; 5103
003773' 102021  000220
003774' 510400  002004                          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>           ; 5104
003775' 431020  005340
```

```
 3217  003776' 510551  070040               MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>         ; 5105
 3218  003777' 001400  015060
 3219  004000' 510651  050000               MWORD   <JMAP,J=5105,D=1>                              ; 5106
 3220  004001' 001000  000040
 3221  004002' 510700  000200               MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>             ; 5107
 3222  004003' 431010  005340
 3223  004004' 511051  102004               MWORD   <JMAP,J=5110,SOA,OR,A=4,D=1,OENA,SELE,MGC=4>   ; 5110
 3224  004005' 431040  005040
 3225  004006' 777777  777777               -1
```

```
                                        ;#*********************************************************************
                                        ;* Test 15 - Fmtr Up Shift 4 (from PLIN) Test - SELRHTNIB=0
                                        ;
                                        ; Description:  Verify that the formatter can shift various  data
                                        ;               patterns 4 bits at a time  while  inserting bytes
                                        ;               from the PLIN into the formatter. The data loaded
                                        ;               into the PLIN initially is 017.
                                        ;
                                        ; Procedure:    KL> Port Clear
                                        ;               KL> Set up data transfer for data patterns
                                        ;
                                        ;               UC> Read a data pattern over the CBUS
                                        ;               UC> Load PLIN with 017
                                        ;
                                        ;               UC> Shift up 4 bits
                                        ;               UC> Read formatter contents and write into EBUF
                                        ;               KL> Read EBUF and verify the data
                                        ;               KL> Start up the ucode to shift again
                                        ;
                                        ;               Repeat the above shifting then verifying EBUF
                                        ;                 contents 10 times, until the data becomes
                                        ;                 all zero.
                                        ;
                                        ;               Do for data patterns:   0's
                                        ;                                       1's
                                        ;                                       525252,,525252
                                        ;                                       000000,,007777
                                        ;                                       000000,,007070
                                        ;                                       000000,,000707
                                        ;
                                        ; Failure:      ---
                                        ;#*********************************************************************

                                        ; Test data

004007' 254 00 0 00 004021'             TSTC15: JRST    TG15                    ; go start test
004010' 100604  000015                          CBUS!NDMP!NDCB!ZCBUS!15         ; test mask
004011' 004231' 013360'                         T15M,,[ASCIZ ^Fmtr Up Shift 4 (from PLIN) Test - SELRHTNIB=0^]
004012' 013263' 012716'                         [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
004013' 000000  004322'                         TSTC16                          ; failure test table
004014' 000000  004633'                         TSTC17                          ;   ...
004015' 000000  005144'                         TSTC20
004016' 000000  005456'                         TSTC21
004017' 000000  005736'                         TSTC22
004020' 777777  777777                          -1
```

```

                                 ; Start test

004021' 201 00 0 00 000000'      TG15:   MOVEI   Z9                      ; get address of module start
004022' 260 17 0 00 003522*              GO      TRACE                   ; handle trace output
004023' 201 01 0 00 004231'              MOVEI   1,T15M                  ; set up microcode address
004024' 260 17 0 00 003524*              GO      TLOAD                   ; load/verify it
004025' 263 17 0 00 000000               RTN                             ; failed - exit test

                                 ; Initialization

004026' 400 15 0 00 000000       TL15:   SETZ    ERFLG,                  ; clear error flag
004027' 260 17 0 00 003527*              GO      IPACLR                  ; clear port
004030' 402 00 0 00 003530*              SETZM   TSTSUB                  ; initialize subtest number
                                         MOVSI   [000000,,000000         ; get address of buffer
                                                  777777,,777777         ;   contents
                                                  525252,,525252
                                                  000000,,007777
                                                  000000,,007070
004031' 205 00 0 00 013330'                       000000,,000707]
004032' 541 00 0 00 003545*              HRRI    BUFF                    ; build the BLT pointer
004033' 251 00 0 00 000000#              BLT     BUFF+5                  ; insert data into buffer

                                 ; First ensure the data transfer is reset

004034' 201 01 0 00 005000               MOVEI   1,5000                  ; get start address
004035' 242 01 0 00 000001               LSH     1,1                     ; position correctly
004036' 260 17 0 00 003670*              GO      LDRAR                   ; load the register
004037' 201 01 0 00 000010               MOVEI   1,MPRUN                 ; get MPRUN bit
004040' 260 17 0 00 003674*              GO      LDCSR                   ; start up the port
004041' 400 01 0 00 000000               SETZ    1,                      ; clear data
004042' 260 17 0 00 004040*              GO      LDCSR                   ; stop the port

                                 ; Now generate CCW list

004043' 200 01 0 00 013210'              MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
004044' 260 17 0 00 003544*              GO      CHINIT                  ; initialize software
004045' 551 01 0 00 004032*              HRRZI   1,BUFF                  ; buffer address
004046' 201 02 0 00 000006               MOVEI   2,6                     ; word count
004047' 400 03 0 00 000000               SETZ    3,                      ; clear zero fill specifier
004050' 260 17 0 00 003550*              GO      GENCCW                  ; generate a CCW list
004051' 201 06 0 00 004074'              MOVEI   6,TS15                  ; get test execute table address

                                 ; Start up the data transfer

004052' 201 01 0 00 001000               MOVEI   1,1000                  ; get start address
004053' 242 01 0 00 000001               LSH     1,1                     ; position correctly
004054' 260 17 0 00 004036*              GO      LDRAR                   ; load the register
004055' 201 01 0 00 000010               MOVEI   1,MPRUN                 ; get MPRUN bit
004056' 260 17 0 00 004042*              GO      LDCSR                   ; start up the port
004057' 201 00 0 00 000001               MOVEI   1                       ; set up delay of 1 ms
004060' 260 17 0 00 003560*              GO      ODELAY                  ; wait a bit
004061' 400 01 0 00 000000               SETZ    1,                      ; clear data
004062' 260 17 0 00 004056*              GO      LDCSR                   ; stop the port
```

```

                                        ; Loop on test execute table entries

004063' 260 17 0 00 003563*     TA15:   GO      TEXEC                           ; execute table entry
004064' 254 00 0 00 004073'             JRST    TX15                            ; end of table
004065' 254 00 0 00 004063'             JRST    TA15                            ; keep looping after call
004066' 474 15 0 00 000000              SETO    ERFLG,                          ; error - EBUF data incorrect

                                        ; Handle error printouts and scope looping

004067' 027 00 0 00 004216'             SCOPER  MA15                            ; print error message
004070' 254 00 0 00 004026'             JRST    TL15                            ; loop on error
004071' 254 00 0 00 004073'             JRST    TX15                            ; altmode exit
004072' 322 15 0 00 004063'             JUMPE   ERFLG,TA15                      ; do next sstep table entry

                                        ; End of test

004073' 263 17 0 00 000000      TX15:   RTN                                     ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

004074' 140000  004150'         TS15:   TTABLE  (TCALL,TS15IN)                  ; initialize stuff
004075' 200000  004153'         TS15L:  TTABLE  (TCALLC,TS15NE)                 ; set up next data pattern

                                        ; Shifted 4 bits

004076' 140000  004201'         TS15A:  TTABLE  (TCALL,TS15NN)                  ; increment shift count
004077' 400000  000100                  TTABLE  (TSSTAR,100)                    ; start up port
004100' 240001  004204'                 TTABLE  (TCHECK,1,TS15RS)               ; check if completed
004101' 340000  000000                  TTABLE  (TEXIT)                         ; exit if error yet

                                        ; Shifted 8 bits

004102' 140000  004201'                 TTABLE  (TCALL,TS15NN)                  ; increment shift count
004103' 400000  000100                  TTABLE  (TSSTAR,100)                    ; start up port
004104' 240001  004205'                 TTABLE  (TCHECK,1,TS15RS+1)             ; check if completed
004105' 340000  000000                  TTABLE  (TEXIT)                         ; exit if error yet

                                        ; Shifted 12 bits

004106' 140000  004201'                 TTABLE  (TCALL,TS15NN)                  ; increment shift count
004107' 400000  000100                  TTABLE  (TSSTAR,100)                    ; start up port
004110' 240001  004206'                 TTABLE  (TCHECK,1,TS15RS+2)             ; check if completed
004111' 340000  000000                  TTABLE  (TEXIT)                         ; exit if error yet

                                        ; Shifted 16 bits

004112' 140000  004201'                 TTABLE  (TCALL,TS15NN)                  ; increment shift count
004113' 400000  000100                  TTABLE  (TSSTAR,100)                    ; start up port
004114' 240001  004207'                 TTABLE  (TCHECK,1,TS15RS+3)             ; check if completed
004115' 340000  000000                  TTABLE  (TEXIT)                         ; exit if error yet

```

```

                                        ; Shifted 20 bits

004116' 140000  004201'                        TTABLE  (TCALL,TS15NN)          ; increment shift count
004117' 400000  000100                         TTABLE  (TSSTAR,100)            ; start up port
004120' 240001  004210'                        TTABLE  (TCHECK,1,TS15RS+4)     ; check if completed
004121' 340000  000000                         TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 24 bits

004122' 140000  004201'                        TTABLE  (TCALL,TS15NN)          ; increment shift count
004123' 400000  000100                         TTABLE  (TSSTAR,100)            ; start up port
004124' 240001  004211'                        TTABLE  (TCHECK,1,TS15RS+5)     ; check if completed
004125' 340000  000000                         TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 28 bits

004126' 140000  004201'                        TTABLE  (TCALL,TS15NN)          ; increment shift count
004127' 400000  000100                         TTABLE  (TSSTAR,100)            ; start up port
004130' 240001  004212'                        TTABLE  (TCHECK,1,TS15RS+6)     ; check if completed
004131' 340000  000000                         TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 32 bits

004132' 140000  004201'                        TTABLE  (TCALL,TS15NN)          ; increment shift count
004133' 400000  000100                         TTABLE  (TSSTAR,100)            ; start up port
004134' 240001  004213'                        TTABLE  (TCHECK,1,TS15RS+7)     ; check if completed
004135' 340000  000000                         TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 36 bits

004136' 140000  004201'                        TTABLE  (TCALL,TS15NN)          ; increment shift count
004137' 400000  000100                         TTABLE  (TSSTAR,100)            ; start up port
004140' 240001  004214'                        TTABLE  (TCHECK,1,TS15RS+10)    ; check if completed
004141' 340000  000000                         TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 40 bits

004142' 140000  004201'                        TTABLE  (TCALL,TS15NN)          ; increment shift count
004143' 400000  000100                         TTABLE  (TSSTAR,100)            ; start up port
004144' 240001  004215'                        TTABLE  (TCHECK,1,TS15RS+11)    ; check if completed
004145' 340000  000000                         TTABLE  (TEXIT)                 ; exit if error yet
004146' 300000  004075'                        TTABLE  (TJRST,TS15L)           ; loop till done
004147' 000000  000000                         TTABLE  (TLAST)

                                        ; Initialize stuff

004150' 201 00 0 00 000000#             TS15IN: MOVEI   BUFF-1                  ; initialize buffer pointer
004151' 202 00 0 00 014017'                    MOVEM   T15BUF#                 ;  ...
004152' 263 17 0 00 000000                     RTN                             ; return
```

```

                                        ; Set up next data pattern

004153' 350 01 0 00 014017'     TS15NE: AOS     1,T15BUF                ; point to next data pattern
004154' 307 01 0 00 000000#             CAIG    1,BUFF+5                ; done yet?
004155' 350 00 0 17 000000              AOS     (P)                     ; no - set up proper return
004156' 200 00 0 01 000000              MOVE    (1)                     ; get data pattern
004157' 201 02 0 00 004204'             MOVEI   2,TS15RS                ; get initial address of EBUF data
004160' 400 01 0 00 000000      TS15NO: SETZ    1,                      ; first shift doesn't include PLIN data
004161' 302 02 0 00 004204'             CAIE    2,TS15RS                ;   so check if first shift and if not
004162' 505 01 0 00 741777              HRLI    1,741777                ;   get initial PLIN data (left 8 bits)
004163' 246 00 0 00 000004              LSHC    4                       ; shift 4 bits
004164' 202 00 0 02 000000              MOVEM   (2)                     ; save it
004165' 350 00 0 00 000002              AOS     2                       ; point to next address
004166' 307 02 0 00 004215'             CAIG    2,TS15RS+11             ; done yet?
004167' 254 00 0 00 004160'             JRST    TS15NO                  ; no - loop till done
004170' 402 00 0 00 014042'             SETZM   TS15NM#                 ; initialize number of shifts

                                ; Read the next word

004171' 201 01 0 00 000200              MOVEI   1,200                   ; get start address
004172' 242 01 0 00 000001              LSH     1,1                     ; position correctly
004173' 260 17 0 00 004054*             GO      LDRAR                   ; load the register
004174' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
004175' 260 17 0 00 004062*             GO      LDCSR                   ; start up the port
004176' 400 01 0 00 000000              SETZ    1,                      ; clear data
004177' 260 17 0 00 004175*             GO      LDCSR                   ; stop the port
004200' 263 17 0 00 000000              RTN                             ; return

                                ; Increment shift count

004201' 201 00 0 00 000004      TS15NN: MOVEI   4                       ; get a 4
004202' 272 00 0 00 014042'             ADDM    TS15NM                  ; add it
004203' 263 17 0 00 000000              RTN                             ; return

004204'                         TS15RS: BLOCK   12                      ; correct EBUF data

                                ; Error messages

004216' 240000  004220'         MA15:   CALL!TXNOT!MA15PN               ; print data pattern ...
004217' 000000000000#                   LAST!CALL!TXALL!TTPNT           ; print test data

004220' 037 00 0 00 013336'     MA15PN: TMSGC   <Data >                 ; get data and
004221' 200 01 0 00 014017'             MOVE    1,T15BUF                ;   print it
004222' 200 00 0 01 000000              MOVE    (1)
004223' 037 13 0 00 000000              PNTHW
004224' 037 00 0 00 013372'             TMSG    < shifted up from PLIN >
004225' 200 00 0 00 014042'             MOVE    TS15NM                  ; get shift count
004226' 037 15 0 00 000000              PNTDEC
004227' 037 00 0 00 013343'             TMSG    <. bits, result in EBUF>
004230' 263 17 0 00 000000              RTN
```

```
3479
3480                                    ; Microcode - Shift 4 bits
3481
3482  004231' 010001  010000            T15M:   MWORD   <ADDR=100,JMAP,J=101,D=1>                       ; 100
3483  004232' 001000  000040
3484  004233' 010100  000120                    MWORD   <CONT,D=1,SELF,MGC=120>                         ; 101
3485  004234' 001000  003340
3486  004235' 010200  000200                    MWORD   <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>              ; 102
3487  004236' 732002  002340
3488  004237' 010351  000000                    MWORD   <JMAP,J=5100,D=1>                               ; 103
3489  004240' 001000  000040
3490
3491                                    ;       Routine to read a word from CBUS into the formatter.
3492
3493  004241' 020002  010000                    MWORD   <ADDR=200,JMAP,J=201>                           ; 200
3494  004242' 000000  000040
3495  004243' 020102  030000                    MWORD   <CJP,J=203,D=1,CENA,CCCA>                       ; 201
3496  004244' 001400  000060
3497  004245' 020202  010000                    MWORD   <JMAP,J=201,D=1>                                ; 202
3498  004246' 001000  000040
3499  004247' 020300  000004                    MWORD   <CONT,D=1,SELC,MGC=4>                           ; 203
3500  004250' 001000  004340
3501  004251' 020400  000200                    MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>              ; 204
3502  004252' 732001  402340
3503  004253' 020500  000000                    MWORD   <JZ>                                            ; 205
3504  004254' 000000  000000
3505
3506                                    ;       Initialization
3507
3508  004255' 100010  010020                    MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>             ; 1000
3509  004256' 000000  002040
3510  004257' 100100  000200                    MWORD   <CONT,D=1,SELC,MGC=200>                         ; 1001
3511  004260' 001000  004340
3512  004261' 100200  310017                    MWORD   <LDCT,J=31,SDO,OR,D=2,SKCN,MGC=017>             ; 1002
3513  004262' 732000  240300
3514  004263' 100310  030000                    MWORD   <RPCT,J=1003,SOA,OR,D=5>                        ; 1003
3515  004264' 435000  000220
3516  004265' 100400  002040                    MWORD   <CONT,SOA,OR,D=1,OENA,SELM,MGC=40>              ; 1004
3517  004266' 431000  002340
3518  004267' 100500  000006                    MWORD   <JZ,SDO,D=1,SELP,MGC=6>                         ; 1005
3519  004270' 701000  001000
3520  004271' 000000  000000                    MWORD   <ADDR=0,JZ>                                     ; 0
3521  004272' 000000  000000
3522
3523                                    ;       Initialization step
3524
3525  004273' 500050  010100                    MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>            ; 5000
3526  004274' 000000  004040
3527  004275' 500100  000020                    MWORD   <JZ,SELM,MGC=20>                                ; 5001
3528  004276' 000000  002000
3529
3530                                    ; Subroutine - Passed - Write 1 to bit 12 of CSR register
3531
3532  004277' 510000  000000                    MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>                ; 5100
3533  004300' 742001  000340
```

```
004301' 510100  000000                MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                        ; 5101
004302' 302001  000740
004303' 510200  260000                MWORD   <LDCT,J=26>                                        ; 5102
004304' 000000  000300
004305' 510351  030000                MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>                 ; 5103
004306' 102021  000220
004307' 510400  002004                MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>              ; 5104
004310' 431020  005340
004311' 510551  070040                MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>             ; 5105
004312' 001400  015060
004313' 510651  050000                MWORD   <JMAP,J=5105,D=1>                                  ; 5106
004314' 001000  000040
004315' 510700  000200                MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>                 ; 5107
004316' 431010  005340
004317' 511051  102004                MWORD   <JMAP,J=5110,D=1,SOA,A=4,OR,OENA,SELE,MGC=4>       ; 5110
004320' 431040  005040
004321' 777777  777777                -1
```

```

                                        ;#*******************************************************************
                                        ;* Test 16 - Fmtr Up Shift 4 (from PLIN) Test - SELRHTNIB=0
                                        ;
                                        ; Description:  Verify that the formatter can shift various  data
                                        ;               patterns up 4 bits at a time while inserting bits
                                        ;               from the PLIN into the formatter. The data loaded
                                        ;               into the PLIN initially is 360.
                                        ;
                                        ; Procedure:    KL> Port Clear
                                        ;               KL> Set up data transfer for data patterns
                                        ;
                                        ;               UC> Read a data pattern over the CBUS
                                        ;               UC> Load PLIN with 360
                                        ;
                                        ;               UC> Shift up 4 bits
                                        ;               UC> Read formatter contents and write into EBUF
                                        ;               KL> Read EBUF and verify the data
                                        ;               KL> Start up the ucode to shift again
                                        ;
                                        ;               Repeat the above shifting then verifying EBUF
                                        ;                 contents 10 times, until the data becomes
                                        ;                 all zero.
                                        ;
                                        ;               Do for data patterns:   0's
                                        ;                                       1's
                                        ;                                       525252,,525252
                                        ;                                       000000,,007777
                                        ;                                       000000,,007070
                                        ;                                       000000,,000707
                                        ;
                                        ; Failure:      ---
                                        ;#*******************************************************************

                                        ; Test data

004322' 254 00 0 00 004334'             TSTC16: JRST    TG16                    ; go start test
004323' 100604  000016                          CBUS!NDMP!NDCB!ZCBUS!16         ; test mask
004324' 004542' 013360'                         T16M,,[ASCIZ ^Fmtr Up Shift 4 (from PLIN) Test - SELRHTNIB=0^]
004325' 013263' 012716'                         [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
004326' 000000  004633'                         TSTC17                          ; failure test table
004327' 000000  005144'                         TSTC20                          ;  ...
004330' 000000  005456'                         TSTC21
004331' 000000  005736'                         TSTC22
004332' 000000  006215'                         TSTC23
004333' 777777  777777                          -1
```

```

                                         ; Start test

004334' 201 00 0 00 000000'   TG16:   MOVEI   Z9                        ; get address of module start
004335' 260 17 0 00 004022*           GO      TRACE                     ; handle trace output
004336' 201 01 0 00 004542'           MOVEI   1,T16M                    ; set up microcode address
004337' 260 17 0 00 004024*           GO      TLOAD                     ; load/verify it
004340' 263 17 0 00 000000            RTN                               ; failed - exit test

                                         ; Initialization

004341' 400 15 0 00 000000    TL16:   SETZ    ERFLG,                    ; clear error flag
004342' 260 17 0 00 004027*           GO      IPACLR                    ; clear port
004343' 402 00 0 00 004030*           SETZM   TSTSUB                    ; initialize subtest number
                                      MOVSI   [000000,,000000           ; get address of buffer
                                               777777,,777777           ;   contents
                                               525252,,525252
                                               000000,,007777
                                               000000,,007070
004344' 205 00 0 00 013330'                    000000,,000707]
004345' 541 00 0 00 004045*           HRRI    BUFF                      ; build the BLT pointer
004346' 251 00 0 00 000000#           BLT     BUFF+5                    ; insert data into buffer

                                         ; First ensure the data transfer is reset

004347' 201 01 0 00 005000            MOVEI   1,5000                    ; get start address
004350' 242 01 0 00 000001            LSH     1,1                       ; position correctly
004351' 260 17 0 00 004173*           GO      LDRAR                     ; load the register
004352' 201 01 0 00 000010            MOVEI   1,MPRUN                   ; get MPRUN bit
004353' 260 17 0 00 004177*           GO      LDCSR                     ; start up the port
004354' 400 01 0 00 000000            SETZ    1,                        ; clear data
004355' 260 17 0 00 004353*           GO      LDCSR                     ; stop the port

                                         ; Now generate CCW list

004356' 200 01 0 00 013210'           MOVE    1,[CBUFSZ,,CBUF]          ; get pointer to control buffer
004357' 260 17 0 00 004044*           GO      CHINIT                    ; initialize software
004360' 551 01 0 00 004345*           HRRZI   1,BUFF                    ; buffer address
004361' 201 02 0 00 000006            MOVEI   2,6                       ; word count
004362' 400 03 0 00 000000            SETZ    3,                        ; clear zero fill specifier
004363' 260 17 0 00 004050*           GO      GENCCW                    ; generate a CCW list
004364' 201 06 0 00 004407'           MOVEI   6,TS16                    ; get test execute table address

                                         ; Start up the data transfer

004365' 201 01 0 00 001000            MOVEI   1,1000                    ; get start address
004366' 242 01 0 00 000001            LSH     1,1                       ; position correctly
004367' 260 17 0 00 004351*           GO      LDRAR                     ; load the register
004370' 201 01 0 00 000010            MOVEI   1,MPRUN                   ; get MPRUN bit
004371' 260 17 0 00 004355*           GO      LDCSR                     ; start up the port
004372' 201 00 0 00 000001            MOVEI   1                         ; set up delay of 1 ms
004373' 260 17 0 00 004060*           GO      ODELAY                    ; wait a bit
004374' 400 01 0 00 000000            SETZ    1,                        ; clear data
004375' 260 17 0 00 004371*           GO      LDCSR                     ; stop the port
```

```

                                        ; Loop on test execute table entries

004376' 260 17 0 00 004063*     TA16:   GO      TEXEC                   ; execute table entry
004377' 254 00 0 00 004406'             JRST    TX16                    ; end of table
004400' 254 00 0 00 004376'             JRST    TA16                    ; keep looping after call
004401' 474 15 0 00 000000              SETO    ERFLG,                  ; error - EBUF data incorrect

                                        ; Handle error printouts and scope looping

004402' 027 00 0 00 004527'             SCOPER  MA16                    ; print error message
004403' 254 00 0 00 004341'             JRST    TL16                    ; loop on error
004404' 254 00 0 00 004406'             JRST    TX16                    ; altmode exit
004405' 322 15 0 00 004376'             JUMPE   ERFLG,TA16              ; do next sstep table entry

                                        ; End of test

004406' 263 17 0 00 000000      TX16:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

004407' 140000  004463'         TS16:   TTABLE  (TCALL,TS16IN)          ; initialize stuff
004410' 200000  004466'         TS16L:  TTABLE  (TCALLC,TS16NE)         ; set up next data pattern

                                        ; Shifted 4 bits

004411' 140000  004512'         TS16A:  TTABLE  (TCALL,TS16NN)          ; increment shift count
004412' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
004413' 240001  004515'                 TTABLE  (TCHECK,1,TS16RS)       ; check if completed
004414' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 8 bits

004415' 140000  004512'                 TTABLE  (TCALL,TS16NN)          ; increment shift count
004416' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
004417' 240001  004516'                 TTABLE  (TCHECK,1,TS16RS+1)     ; check if completed
004420' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 12 bits

004421' 140000  004512'                 TTABLE  (TCALL,TS16NN)          ; increment shift count
004422' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
004423' 240001  004517'                 TTABLE  (TCHECK,1,TS16RS+2)     ; check if completed
004424' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 16 bits

004425' 140000  004512'                 TTABLE  (TCALL,TS16NN)          ; increment shift count
004426' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
004427' 240001  004520'                 TTABLE  (TCHECK,1,TS16RS+3)     ; check if completed
004430' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

```

```

                                  ; Shifted 20 bits

004431' 140000  004512'                   TTABLE  (TCALL,TS16NN)            ; increment shift count
004432' 400000  000100                    TTABLE  (TSSTAR,100)              ; start up port
004433' 240001  004521'                   TTABLE  (TCHECK,1,TS16RS+4)       ; check if completed
004434' 340000  000000                    TTABLE  (TEXIT)                   ; exit if error yet

                                  ; Shifted 24 bits

004435' 140000  004512'                   TTABLE  (TCALL,TS16NN)            ; increment shift count
004436' 400000  000100                    TTABLE  (TSSTAR,100)              ; start up port
004437' 240001  004522'                   TTABLE  (TCHECK,1,TS16RS+5)       ; check if completed
004440' 340000  000000                    TTABLE  (TEXIT)                   ; exit if error yet

                                  ; Shifted 28 bits

004441' 140000  004512'                   TTABLE  (TCALL,TS16NN)            ; increment shift count
004442' 400000  000100                    TTABLE  (TSSTAR,100)              ; start up port
004443' 240001  004523'                   TTABLE  (TCHECK,1,TS16RS+6)       ; check if completed
004444' 340000  000000                    TTABLE  (TEXIT)                   ; exit if error yet

                                  ; Shifted 32 bits

004445' 140000  004512'                   TTABLE  (TCALL,TS16NN)            ; increment shift count
004446' 400000  000100                    TTABLE  (TSSTAR,100)              ; start up port
004447' 240001  004524'                   TTABLE  (TCHECK,1,TS16RS+7)       ; check if completed
004450' 340000  000000                    TTABLE  (TEXIT)                   ; exit if error yet

                                  ; Shifted 36 bits

004451' 140000  004512'                   TTABLE  (TCALL,TS16NN)            ; increment shift count
004452' 400000  000100                    TTABLE  (TSSTAR,100)              ; start up port
004453' 240001  004525'                   TTABLE  (TCHECK,1,TS16RS+10)      ; check if completed
004454' 340000  000000                    TTABLE  (TEXIT)                   ; exit if error yet

                                  ; Shifted 40 bits

004455' 140000  004512'                   TTABLE  (TCALL,TS16NN)            ; increment shift count
004456' 400000  000100                    TTABLE  (TSSTAR,100)              ; start up port
004457' 240001  004526'                   TTABLE  (TCHECK,1,TS16RS+11)      ; check if completed
004460' 340000  000000                    TTABLE  (TEXIT)                   ; exit if error yet
004461' 300000  004410'                   TTABLE  (TJRST,TS16L)             ; loop till done
004462' 000000  000000                    TTABLE  (TLAST)

                                  ; Initialize stuff

004463' 201 00 0 00 000000#       TS16IN: MOVEI   BUFF-1                    ; initialize buffer pointer
004464' 202 00 0 00 014020'               MOVEM   T16BUF#                   ;  ...
004465' 263 17 0 00 000000                RTN                               ; return
```

```
                                        ; Set up next data pattern

004466' 350 01 0 00 014020'             TS16NE: AOS     1,T16BUF                ; point to next data pattern
004467' 307 01 0 00 000000#                     CAIG    1,BUFF+5                ; done yet?
004470' 350 00 0 17 000000                      AOS     (P)                     ; no - set up proper return
004471' 200 00 0 01 000000                      MOVE    (1)                     ; get data pattern
004472' 201 02 0 00 004515'                     MOVEI   2,TS16RS                ; get initial address of EBUF data
004473' 505 01 0 00 037777              TS16NO: HRLI    1,37777                 ; get initial PLIN data (left 8 bits)
004474' 246 00 0 00 000004                      LSHC    4                       ; shift 4 bits
004475' 202 00 0 02 000000                      MOVEM   (2)                     ; save it
004476' 350 00 0 00 000002                      AOS     2                       ; point to next address
004477' 307 02 0 00 004526'                     CAIG    2,TS16RS+11             ; done yet?
004500' 254 00 0 00 004473'                     JRST    TS16NO                  ; no - loop till done
004501' 402 00 0 00 014043'                     SETZM   TS16NM#                 ; initialize number of shifts

                                        ; Read the next word

004502' 201 01 0 00 000200                      MOVEI   1,200                   ; get start address
004503' 242 01 0 00 000001                      LSH     1,1                     ; position correctly
004504' 260 17 0 00 004367*                     GO      LDRAR                   ; load the register
004505' 201 01 0 00 000010                      MOVEI   1,MPRUN                 ; get MPRUN bit
004506' 260 17 0 00 004375*                     GO      LDCSR                   ; start up the port
004507' 400 01 0 00 000000                      SETZ    1,                      ; clear data
004510' 260 17 0 00 004506*                     GO      LDCSR                   ; stop the port
004511' 263 17 0 00 000000                      RTN                             ; return

                                        ; Increment shift count

004512' 201 00 0 00 000004              TS16NN: MOVEI   4                       ; get a 4
004513' 272 00 0 00 014043'                     ADDM    TS16NM                  ; add it
004514' 263 17 0 00 000000                      RTN                             ; return

004515'                                 TS16RS: BLOCK   12                      ; correct EBUF data

                                        ; Error messages

004527' 240000  004531'                 MA16:   CALL!TXNOT!MA16PN               ; print data pattern ...
004530' 000000000000#                           LAST!CALL!TXALL!TTPNT           ; print test data

004531' 037 00 0 00 013336'             MA16PN: TMSGC   <Data >                 ; get data and
004532' 200 01 0 00 014020'                     MOVE    1,T16BUF                ;   print it
004533' 200 00 0 01 000000                      MOVE    (1)
004534' 037 13 0 00 000000                      PNTHW
004535' 037 00 0 00 013372'                     TMSG    < shifted up from PLIN >
004536' 200 00 0 00 014043'                     MOVE    TS16NM                  ; get shift count
004537' 037 15 0 00 000000                      PNTDEC
004540' 037 00 0 00 013343'                     TMSG    <. bits, result in EBUF>
004541' 263 17 0 00 000000                      RTN
```

```

                        ; Microcode - Shift 4 bits

004542' 010001  010000  T16M:   MWORD   <ADDR=100,JMAP,J=101,D=1>               ; 100
004543' 001000  000040
004544' 010100  000120          MWORD   <CONT,D=1,SELF,MGC=120>                 ; 101
004545' 001000  003340
004546' 010200  000200          MWORD   <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>      ; 102
004547' 732002  002340
004550' 010351  000000          MWORD   <JMAP,J=5100,D=1>                       ; 103
004551' 001000  000040

                        ;       Routine to read a word from CBUS into the formatter.

004552' 020002  010000          MWORD   <ADDR=200,JMAP,J=201>                   ; 200
004553' 000000  000040
004554' 020102  030000          MWORD   <CJP,J=203,D=1,CENA,CCCA>               ; 201
004555' 001400  000060
004556' 020202  010000          MWORD   <JMAP,J=201,D=1>                        ; 202
004557' 001000  000040
004560' 020300  000004          MWORD   <CONT,D=1,SELC,MGC=4>                   ; 203
004561' 001000  004340
004562' 020400  000200          MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>      ; 204
004563' 732001  402340
004564' 020500  000000          MWORD   <JZ>                                    ; 205
004565' 000000  000000

                        ;       Initialization

004566' 100010  010020          MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>     ; 1000
004567' 000000  002040
004570' 100100  000200          MWORD   <CONT,D=1,SELC,MGC=200>                 ; 1001
004571' 001000  004340
004572' 100200  310360          MWORD   <LDCT,J=31,SDO,OR,D=2,SKCN,MGC=360>     ; 1002
004573' 732000  240300
004574' 100310  030000          MWORD   <RPCT,J=1003,SOA,OR,D=5>                ; 1003
004575' 435000  000220
004576' 100400  002040          MWORD   <CONT,SOA,OR,D=1,OENA,SELM,MGC=40>      ; 1004
004577' 431000  002340
004600' 100500  000006          MWORD   <JZ,SDO,D=1,SELP,MGC=6>                 ; 1005
004601' 701000  001000
004602' 000000  000000          MWORD   <ADDR=0,JZ>                             ; 0
004603' 000000  000000

                        ;       Initialization step

004604' 500050  010100          MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>    ; 5000
004605' 000000  004040
004606' 500100  000020          MWORD   <JZ,SELM,MGC=20>                        ; 5001
004607' 000000  002000

                        ; Subroutine - Passed - Write 1 to bit 12 of CSR register

004610' 510000  000000          MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>        ; 5100
004611' 742001  000340
```

```
 3857  004612' 510100  000000            MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                        ; 5101
 3858  004613' 302001  000740
 3859  004614' 510200  260000            MWORD   <LDCT,J=26>                                        ; 5102
 3860  004615' 000000  000300
 3861  004616' 510351  030000            MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>                 ; 5103
 3862  004617' 102021  000220
 3863  004620' 510400  002004            MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>              ; 5104
 3864  004621' 431020  005340
 3865  004622' 510551  070040            MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>             ; 5105
 3866  004623' 001400  015060
 3867  004624' 510651  050000            MWORD   <JMAP,J=5105,D=1>                                  ; 5106
 3868  004625' 001000  000040
 3869  004626' 510700  000200            MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>                 ; 5107
 3870  004627' 431010  005340
 3871  004630' 511051  102004            MWORD   <JMAP,J=5110,SOA,A=4,OR,D=1,OENA,SELE,MGC=4>       ; 5110
 3872  004631' 431040  005040
 3873  004632' 777777  777777            -1
```

```

                                ;#*********************************************************************
                                ;* Test 17 - Fmtr Up Shift 4 (from PLIN) Test - SELRHTNIB=1
                                ;
                                ; Description:  Verify that the formatter can shift various  data
                                ;               patterns up 4 bits at a time while inserting bits
                                ;               from the PLIN into the formatter. The data loaded
                                ;               into the PLIN initially is 017.
                                ;
                                ; Procedure:    KL> Port Clear
                                ;               KL> Set up data transfer for data patterns
                                ;
                                ;               UC> Read a data pattern over the CBUS
                                ;               UC> Load PLIN with 017
                                ;
                                ;               UC> Shift up 4 bits
                                ;               UC> Read formatter contents and write into EBUF
                                ;               KL> Read EBUF and verify the data
                                ;               KL> Start up the ucode to shift again
                                ;
                                ;               Repeat the above shifting then verifying EBUF
                                ;                 contents 10 times, until the data becomes
                                ;                 all zero.
                                ;
                                ;               Do for data patterns:   0's
                                ;                                       1's
                                ;                                       525252,,525252
                                ;                                       000000,,007777
                                ;                                       000000,,007070
                                ;                                       000000,,000707
                                ;
                                ; Failure:      ---
                                ;#*********************************************************************

                                ; Test data

004633' 254 00 0 00 004645'     TSTC17: JRST    TG17                    ; go start test
004634' 100604  000017                  CBUS!NDMP!NDCB!ZCBUS!17         ; test mask
004635' 005053' 013377'                 T17M,,[ASCIZ ^Fmtr Up Shift 4 (from PLIN) Test - SELRHTNIB=1^]
004636' 013263' 012716'                 [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
004637' 000000  005144'                 TSTC20                          ; failure test table
004640' 000000  005456'                 TSTC21                          ;   ...
004641' 000000  005736'                 TSTC22
004642' 000000  006215'                 TSTC23
004643' 000000  006473'                 TSTC24
004644' 777777  777777                  -1
```

```

                                  ; Start test

004645' 201 00 0 00 000000'       TG17:   MOVEI   Z9                              ; get address of module start
004646' 260 17 0 00 004335*               GO      TRACE                           ; handle trace output
004647' 201 01 0 00 005053'               MOVEI   1,T17M                          ; set up microcode address
004650' 260 17 0 00 004337*               GO      TLOAD                           ; load/verify it
004651' 263 17 0 00 000000                RTN                                     ; failed - exit test

                                  ; Initialization

004652' 400 15 0 00 000000        TL17:   SETZ    ERFLG,                          ; clear error flag
004653' 260 17 0 00 004342*               GO      IPACLR                          ; clear port
004654' 402 00 0 00 004343*               SETZM   TSTSUB                          ; initialize subtest number
                                          MOVSI   [000000,,000000                 ; get address of buffer
                                                   777777,,777777                 ;   contents
                                                   525252,,525252
                                                   000000,,007777
                                                   000000,,007070
004655' 205 00 0 00 013330'                        000000,,000707]
004656' 541 00 0 00 004360*               HRRI    BUFF                            ; build the BLT pointer
004657' 251 00 0 00 000000#               BLT     BUFF+5                          ; insert data into buffer

                                  ; First ensure the data transfer is reset

004660' 201 01 0 00 005000                MOVEI   1,5000                          ; get start address
004661' 242 01 0 00 000001                LSH     1,1                             ; position correctly
004662' 260 17 0 00 004504*               GO      LDRAR                           ; load the register
004663' 201 01 0 00 000010                MOVEI   1,MPRUN                         ; get MPRUN bit
004664' 260 17 0 00 004510*               GO      LDCSR                           ; start up the port
004665' 400 01 0 00 000000                SETZ    1,                              ; clear data
004666' 260 17 0 00 004664*               GO      LDCSR                           ; stop the port

                                  ; Now generate CCW list

004667' 200 01 0 00 013210'               MOVE    1,[CBUFSZ,,CBUF]                ; get pointer to control buffer
004670' 260 17 0 00 004357*               GO      CHINIT                          ; initialize software
004671' 551 01 0 00 004656*               HRRZI   1,BUFF                          ; buffer address
004672' 201 02 0 00 000006                MOVEI   2,6                             ; word count
004673' 400 03 0 00 000000                SETZ    3,                              ; clear zero fill specifier
004674' 260 17 0 00 004363*               GO      GENCCW                          ; generate a CCW list
004675' 201 06 0 00 004720'               MOVEI   6,TS17                          ; get test execute table address

                                  ; Start up the data transfer

004676' 201 01 0 00 001000                MOVEI   1,1000                          ; get start address
004677' 242 01 0 00 000001                LSH     1,1                             ; position correctly
004700' 260 17 0 00 004662*               GO      LDRAR                           ; load the register
004701' 201 01 0 00 000010                MOVEI   1,MPRUN                         ; get MPRUN bit
004702' 260 17 0 00 004666*               GO      LDCSR                           ; start up the port
004703' 201 00 0 00 000001                MOVEI   1                               ; set up delay of 1 ms
004704' 260 17 0 00 004373*               GO      ODELAY                          ; wait a bit
004705' 400 01 0 00 000000                SETZ    1,                              ; clear data
004706' 260 17 0 00 004702*               GO      LDCSR                           ; stop the port
```

```

                                    ; Loop on test execute table entries

004707' 260 17 0 00 004376*         TA17:   GO      TEXEC                   ; execute table entry
004710' 254 00 0 00 004717'                 JRST    TX17                    ; end of table
004711' 254 00 0 00 004707'                 JRST    TA17                    ; keep looping after call
004712' 474 15 0 00 000000                  SETO    ERFLG,                  ; error - EBUF data incorrect

                                    ; Handle error printouts and scope looping

004713' 027 00 0 00 005040'                 SCOPER  MA17                    ; print error message
004714' 254 00 0 00 004652'                 JRST    TL17                    ; loop on error
004715' 254 00 0 00 004717'                 JRST    TX17                    ; altmode exit
004716' 322 15 0 00 004707'                 JUMPE   ERFLG,TA17              ; do next sstep table entry

                                    ; End of test

004717' 263 17 0 00 000000          TX17:   RTN                             ; return

                                    ; Test Execute Table, as:  (CMD,parameters)

004720' 140000  004774'             TS17:   TTABLE  (TCALL,TS17IN)          ; initialize stuff
004721' 200000  004777'             TS17L:  TTABLE  (TCALLC,TS17NE)         ; set up next data pattern

                                    ; Shifted 4 bits

004722' 140000  005023'             TS17A:  TTABLE  (TCALL,TS17NN)          ; increment shift count
004723' 400000  000100                      TTABLE  (TSSTAR,100)            ; start up port
004724' 240001  005026'                     TTABLE  (TCHECK,1,TS17RS)       ; check if completed
004725' 340000  000000                      TTABLE  (TEXIT)                 ; exit if error yet

                                    ; Shifted 8 bits

004726' 140000  005023'                     TTABLE  (TCALL,TS17NN)          ; increment shift count
004727' 400000  000100                      TTABLE  (TSSTAR,100)            ; start up port
004730' 240001  005027'                     TTABLE  (TCHECK,1,TS17RS+1)     ; check if completed
004731' 340000  000000                      TTABLE  (TEXIT)                 ; exit if error yet

                                    ; Shifted 12 bits

004732' 140000  005023'                     TTABLE  (TCALL,TS17NN)          ; increment shift count
004733' 400000  000100                      TTABLE  (TSSTAR,100)            ; start up port
004734' 240001  005030'                     TTABLE  (TCHECK,1,TS17RS+2)     ; check if completed
004735' 340000  000000                      TTABLE  (TEXIT)                 ; exit if error yet

                                    ; Shifted 16 bits

004736' 140000  005023'                     TTABLE  (TCALL,TS17NN)          ; increment shift count
004737' 400000  000100                      TTABLE  (TSSTAR,100)            ; start up port
004740' 240001  005031'                     TTABLE  (TCHECK,1,TS17RS+3)     ; check if completed
004741' 340000  000000                      TTABLE  (TEXIT)                 ; exit if error yet

```

```

                                        ; Shifted 20 bits

004742' 140000  005023'                         TTABLE  (TCALL,TS17NN)          ; increment shift count
004743' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
004744' 240001  005032'                         TTABLE  (TCHECK,1,TS17RS+4)     ; check if completed
004745' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 24 bits

004746' 140000  005023'                         TTABLE  (TCALL,TS17NN)          ; increment shift count
004747' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
004750' 240001  005033'                         TTABLE  (TCHECK,1,TS17RS+5)     ; check if completed
004751' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 28 bits

004752' 140000  005023'                         TTABLE  (TCALL,TS17NN)          ; increment shift count
004753' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
004754' 240001  005034'                         TTABLE  (TCHECK,1,TS17RS+6)     ; check if completed
004755' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 32 bits

004756' 140000  005023'                         TTABLE  (TCALL,TS17NN)          ; increment shift count
004757' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
004760' 240001  005035'                         TTABLE  (TCHECK,1,TS17RS+7)     ; check if completed
004761' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 36 bits

004762' 140000  005023'                         TTABLE  (TCALL,TS17NN)          ; increment shift count
004763' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
004764' 240001  005036'                         TTABLE  (TCHECK,1,TS17RS+10)    ; check if completed
004765' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 40 bits

004766' 140000  005023'                         TTABLE  (TCALL,TS17NN)          ; increment shift count
004767' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
004770' 240001  005037'                         TTABLE  (TCHECK,1,TS17RS+11)    ; check if completed
004771' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet
004772' 300000  004721'                         TTABLE  (TJRST,TS17L)           ; loop till done
004773' 000000  000000                          TTABLE  (TLAST)

                                        ; Initialize stuff

004774' 201 00 0 00 000000#             TS17IN: MOVEI   BUFF-1                  ; initialize buffer pointer
004775' 202 00 0 00 014021'                     MOVEM   T17BUF#                 ;  ...
004776' 263 17 0 00 000000                      RTN                             ; return
```

```

                                        ; Set up next data pattern

004777' 350 01 0 00 014021'         TS17NE: AOS     1,T17BUF                ; point to next data pattern
005000' 307 01 0 00 000000#                 CAIG    1,BUFF+5                ; done yet?
005001' 350 00 0 17 000000                  AOS     (P)                     ; no - set up proper return
005002' 200 00 0 01 000000                  MOVE    (1)                     ; get data pattern
005003' 201 02 0 00 005026'                 MOVEI   2,TS17RS                ; get initial address of EBUF data
005004' 505 01 0 00 037777          TS17NO: HRLI    1,37777                 ; get initial PLIN data (left 8 bits)
005005' 246 00 0 00 000004                  LSHC    4                       ; shift 4 bits
005006' 202 00 0 02 000000                  MOVEM   (2)                     ; save it
005007' 350 00 0 00 000002                  AOS     2                       ; point to next address
005010' 307 02 0 00 005037'                 CAIG    2,TS17RS+11             ; done yet?
005011' 254 00 0 00 005004'                 JRST    TS17NO                  ; no - loop till done
005012' 402 00 0 00 014044'                 SETZM   TS17NM#                 ; initialize number of shifts

                                        ; Read the next word

005013' 201 01 0 00 000200                  MOVEI   1,200                   ; get start address
005014' 242 01 0 00 000001                  LSH     1,1                     ; position correctly
005015' 260 17 0 00 004700*                 GO      LDRAR                   ; load the register
005016' 201 01 0 00 000010                  MOVEI   1,MPRUN                 ; get MPRUN bit
005017' 260 17 0 00 004706*                 GO      LDCSR                   ; start up the port
005020' 400 01 0 00 000000                  SETZ    1,                      ; clear data
005021' 260 17 0 00 005017*                 GO      LDCSR                   ; stop the port
005022' 263 17 0 00 000000                  RTN                             ; return

                                        ; Increment shift count

005023' 201 00 0 00 000004          TS17NN: MOVEI   4                       ; get a 4
005024' 272 00 0 00 014044'                 ADDM    TS17NM                  ; add it
005025' 263 17 0 00 000000                  RTN                             ; return

005026'                             TS17RS: BLOCK   12                      ; correct EBUF data

                                        ; Error messages

005040' 240000  005042'             MA17:   CALL!TXNOT!MA17PN               ; print data pattern ...
005041' 000000000000#                       LAST!CALL!TXALL!TTPNT           ; print test data

005042' 037 00 0 00 013336'         MA17PN: TMSGC   <Data >                 ; get data and
005043' 200 01 0 00 014021'                 MOVE    1,T17BUF                ;   print it
005044' 200 00 0 01 000000                  MOVE    (1)
005045' 037 13 0 00 000000                  PNTHW
005046' 037 00 0 00 013372'                 TMSG    < shifted up from PLIN >
005047' 200 00 0 00 014044'                 MOVE    TS17NM                  ; get shift count
005050' 037 15 0 00 000000                  PNTDEC
005051' 037 00 0 00 013343'                 TMSG    <. bits, result in EBUF>
005052' 263 17 0 00 000000                  RTN
```

```

                                        ; Microcode - Shift 4 bits

005053' 010001  010000                  T17M:   MWORD   <ADDR=100,JMAP,J=101,D=1>                       ; 100
005054' 001000  000040
005055' 010100  000122                          MWORD   <CONT,D=1,SELF,MGC=122>                         ; 101
005056' 001000  003340
005057' 010200  000200                          MWORD   <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>              ; 102
005060' 732002  002340
005061' 010351  000000                          MWORD   <JMAP,J=5100,D=1>                               ; 103
005062' 001000  000040

                                        ;       Routine to read a word from CBUS into the formatter.

005063' 020002  010000                          MWORD   <ADDR=200,JMAP,J=201>                           ; 200
005064' 000000  000040
005065' 020102  030000                          MWORD   <CJP,J=203,D=1,CENA,CCCA>                       ; 201
005066' 001400  000060
005067' 020202  010000                          MWORD   <JMAP,J=201,D=1>                                ; 202
005070' 001000  000040
005071' 020300  000004                          MWORD   <CONT,D=1,SELC,MGC=4>                           ; 203
005072' 001000  004340
005073' 020400  000200                          MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>              ; 204
005074' 732001  402340
005075' 020500  000000                          MWORD   <JZ>                                            ; 205
005076' 000000  000000

                                        ;       Initialization

005077' 100010  010020                          MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>             ; 1000
005100' 000000  002040
005101' 100100  000200                          MWORD   <CONT,D=1,SELC,MGC=200>                         ; 1001
005102' 001000  004340
005103' 100200  310017                          MWORD   <LDCT,J=31,SDO,OR,D=2,SKCN,MGC=017>             ; 1002
005104' 732000  240300
005105' 100310  030000                          MWORD   <RPCT,J=1003,SOA,OR,D=5>                        ; 1003
005106' 435000  000220
005107' 100400  002040                          MWORD   <CONT,SOA,OR,D=1,OENA,SELM,MGC=40>              ; 1004
005110' 431000  002340
005111' 100500  000006                          MWORD   <JZ,SDO,D=1,SELP,MGC=6>                         ; 1005
005112' 701000  001000
005113' 000000  000000                          MWORD   <ADDR=0,JZ>                                     ; 0
005114' 000000  000000

                                        ;       Initialization step

005115' 500050  010100                          MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>            ; 5000
005116' 000000  004040
005117' 500100  000020                          MWORD   <JZ,SELM,MGC=20>                                ; 5001
005120' 000000  002000

                                        ; Subroutine - Passed - Write 1 to bit 12 of CSR register

005121' 510000  000000                          MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>                ; 5100
005122' 742001  000340
```

```
005123' 510100  000000          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                         ; 5101
005124' 302001  000740
005125' 510200  260000          MWORD   <LDCT,J=26>                                         ; 5102
005126' 000000  000300
005127' 510351  030000          MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>                  ; 5103
005130' 102021  000220
005131' 510400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>               ; 5104
005132' 431020  005340
005133' 510551  070040          MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>              ; 5105
005134' 001400  015060
005135' 510651  050000          MWORD   <JMAP,J=5105,D=1>                                   ; 5106
005136' 001000  000040
005137' 510700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>                  ; 5107
005140' 431010  005340
005141' 511051  102004          MWORD   <JMAP,J=5110,SOA,A=4,OR,D=1,OENA,SELE,MGC=4>        ; 5110
005142' 431040  005040
005143' 777777  777777          -1
```

```
                                        ;#*****************************************************************
                                        ;* Test 20 - Fmtr Up Shift 4 (from PLIN) Test - SELRHTNIB=1
                                        ;
                                        ; Description:  Verify that the formatter can shift various  data
                                        ;               patterns up 4 bits at a time while inserting bits
                                        ;               from the PLIN into the formatter. The data loaded
                                        ;               into the PLIN initially is 360.
                                        ;
                                        ; Procedure:    KL> Port Clear
                                        ;               KL> Set up data transfer for data patterns
                                        ;
                                        ;               UC> Read a data pattern over the CBUS
                                        ;               UC> Load PLIN with 360
                                        ;
                                        ;               UC> Shift up 4 bits
                                        ;               UC> Read formatter contents and write into EBUF
                                        ;               KL> Read EBUF and verify the data
                                        ;               KL> Start up the ucode to shift again
                                        ;
                                        ;               Repeat the above shifting then verifying EBUF
                                        ;                 contents 10 times, until the data becomes
                                        ;                 all zero.
                                        ;
                                        ;               Do for data patterns:   0's
                                        ;                                       1's
                                        ;                                       525252,,525252
                                        ;                                       000000,,007777
                                        ;                                       000000,,007070
                                        ;                                       000000,,000707
                                        ;
                                        ; Failure:      ---
                                        ;#*****************************************************************

                                        ; Test data

005144' 254 00 0 00 005155'             TST20:  JRST    TG20                    ; go start test
005145' 100604  000020                          CBUS!NDMP!NDCB!ZCBUS!20         ; test mask
005146' 005365' 013377'                         T20M,,[ASCIZ ^Fmtr Up Shift 4 (from PLIN) Test - SELRHTNIB=1^]
005147' 013263' 012716'                         [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
005150' 000000  005456'                         TSTC21                          ; failure test table
005151' 000000  005736'                         TSTC22                          ;   ...
005152' 000000  006215'                         TSTC23
005153' 000000  006473'                         TSTC24
005154' 777777  777777                          -1
```

```

                                        ; Start test

005155' 201 00 0 00 000000'             TG20:   MOVEI   Z9                      ; get address of module start
005156' 260 17 0 00 004646*                     GO      TRACE                   ; handle trace output
005157' 201 01 0 00 005365'                     MOVEI   1,T20M                  ; set up microcode address
005160' 260 17 0 00 004650*                     GO      TLOAD                   ; load/verify it
005161' 263 17 0 00 000000                      RTN                             ; failed - exit test

                                        ; Initialization

005162' 400 15 0 00 000000              TL20:   SETZ    ERFLG,                  ; clear error flag
005163' 260 17 0 00 004653*                     GO      IPACLR                  ; clear port
005164' 402 00 0 00 004654*                     SETZM   TSTSUB                  ; initialize subtest number
                                                MOVSI   [000000,,000000         ; get address of buffer
                                                         777777,,777777         ;   contents
                                                         525252,,525252
                                                         000000,,007777
                                                         000000,,007070
005165' 205 00 0 00 013330'                              000000,,000707]
005166' 541 00 0 00 004671*                     HRRI    BUFF                    ; build the BLT pointer
005167' 251 00 0 00 000000#                     BLT     BUFF+5                  ; insert data into buffer

                                        ; First ensure the data transfer is reset

005170' 201 01 0 00 005000                      MOVEI   1,5000                  ; get start address
005171' 242 01 0 00 000001                      LSH     1,1                     ; position correctly
005172' 260 17 0 00 005015*                     GO      LDRAR                   ; load the register
005173' 201 01 0 00 000010                      MOVEI   1,MPRUN                 ; get MPRUN bit
005174' 260 17 0 00 005021*                     GO      LDCSR                   ; start up the port
005175' 400 01 0 00 000000                      SETZ    1,                      ; clear data
005176' 260 17 0 00 005174*                     GO      LDCSR                   ; stop the port

                                        ; Now generate CCW list

005177' 200 01 0 00 013210'                     MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
005200' 260 17 0 00 004670*                     GO      CHINIT                  ; initialize software
005201' 551 01 0 00 005166*                     HRRZI   1,BUFF                  ; buffer address
005202' 201 02 0 00 000006                      MOVEI   2,6                     ; word count
005203' 400 03 0 00 000000                      SETZ    3,                      ; clear zero fill specifier
005204' 260 17 0 00 004674*                     GO      GENCCW                  ; generate a CCW list
005205' 201 06 0 00 005230'                     MOVEI   6,TS20                  ; get test execute table address

                                        ; Start up the data transfer

005206' 201 01 0 00 001000                      MOVEI   1,1000                  ; get start address
005207' 242 01 0 00 000001                      LSH     1,1                     ; position correctly
005210' 260 17 0 00 005172*                     GO      LDRAR                   ; load the register
005211' 201 01 0 00 000010                      MOVEI   1,MPRUN                 ; get MPRUN bit
005212' 260 17 0 00 005176*                     GO      LDCSR                   ; start up the port
005213' 201 00 0 00 000001                      MOVEI   1                       ; set up delay of 1 ms
005214' 260 17 0 00 004704*                     GO      ODELAY                  ; wait a bit
005215' 400 01 0 00 000000                      SETZ    1,                      ; clear data
005216' 260 17 0 00 005212*                     GO      LDCSR                   ; stop the port
```

```

                                        ; Loop on test execute table entries

005217' 260 17 0 00 004707*             TA20:   GO      TEXEC                   ; execute table entry
005220' 254 00 0 00 005227'                     JRST    TX20                    ; end of table
005221' 254 00 0 00 005217'                     JRST    TA20                    ; keep looping after call
005222' 474 15 0 00 000000                      SETO    ERFLG,                  ; error - EBUF data incorrect

                                        ; Handle error printouts and scope looping

005223' 027 00 0 00 005352'                     SCOPER  MA20                    ; print error message
005224' 254 00 0 00 005162'                     JRST    TL20                    ; loop on error
005225' 254 00 0 00 005227'                     JRST    TX20                    ; altmode exit
005226' 322 15 0 00 005217'                     JUMPE   ERFLG,TA20              ; do next sstep table entry

                                        ; End of test

005227' 263 17 0 00 000000              TX20:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

005230' 140000  005304'                 TS20:   TTABLE  (TCALL,TS20IN)          ; initialize stuff
005231' 200000  005307'                 TS20L:  TTABLE  (TCALLC,TS20NE)         ; set up next data pattern

                                        ; Shifted 4 bits

005232' 140000  005335'                 TS20A:  TTABLE  (TCALL,TS20NN)          ; increment shift count
005233' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
005234' 240001  005340'                         TTABLE  (TCHECK,1,TS20RS)       ; check if completed
005235' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 8 bits

005236' 140000  005335'                         TTABLE  (TCALL,TS20NN)          ; increment shift count
005237' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
005240' 240001  005341'                         TTABLE  (TCHECK,1,TS20RS+1)     ; check if completed
005241' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 12 bits

005242' 140000  005335'                         TTABLE  (TCALL,TS20NN)          ; increment shift count
005243' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
005244' 240001  005342'                         TTABLE  (TCHECK,1,TS20RS+2)     ; check if completed
005245' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 16 bits

005246' 140000  005335'                         TTABLE  (TCALL,TS20NN)          ; increment shift count
005247' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
005250' 240001  005343'                         TTABLE  (TCHECK,1,TS20RS+3)     ; check if completed
005251' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

```

```

                                 ; Shifted 20 bits

005252' 140000  005335'                  TTABLE  (TCALL,TS20NN)          ; increment shift count
005253' 400000  000100                   TTABLE  (TSSTAR,100)            ; start up port
005254' 240001  005344'                  TTABLE  (TCHECK,1,TS20RS+4)     ; check if completed
005255' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

                                 ; Shifted 24 bits

005256' 140000  005335'                  TTABLE  (TCALL,TS20NN)          ; increment shift count
005257' 400000  000100                   TTABLE  (TSSTAR,100)            ; start up port
005260' 240001  005345'                  TTABLE  (TCHECK,1,TS20RS+5)     ; check if completed
005261' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

                                 ; Shifted 28 bits

005262' 140000  005335'                  TTABLE  (TCALL,TS20NN)          ; increment shift count
005263' 400000  000100                   TTABLE  (TSSTAR,100)            ; start up port
005264' 240001  005346'                  TTABLE  (TCHECK,1,TS20RS+6)     ; check if completed
005265' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

                                 ; Shifted 32 bits

005266' 140000  005335'                  TTABLE  (TCALL,TS20NN)          ; increment shift count
005267' 400000  000100                   TTABLE  (TSSTAR,100)            ; start up port
005270' 240001  005347'                  TTABLE  (TCHECK,1,TS20RS+7)     ; check if completed
005271' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

                                 ; Shifted 36 bits

005272' 140000  005335'                  TTABLE  (TCALL,TS20NN)          ; increment shift count
005273' 400000  000100                   TTABLE  (TSSTAR,100)            ; start up port
005274' 240001  005350'                  TTABLE  (TCHECK,1,TS20RS+10)    ; check if completed
005275' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

                                 ; Shifted 40 bits

005276' 140000  005335'                  TTABLE  (TCALL,TS20NN)          ; increment shift count
005277' 400000  000100                   TTABLE  (TSSTAR,100)            ; start up port
005300' 240001  005351'                  TTABLE  (TCHECK,1,TS20RS+11)    ; check if completed
005301' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet
005302' 300000  005231'                  TTABLE  (TJRST,TS20L)           ; loop till done
005303' 000000  000000                   TTABLE  (TLAST)

                                 ; Initialize stuff

005304' 201 00 0 00 000000#      TS20IN: MOVEI   BUFF-1                  ; initialize buffer pointer
005305' 202 00 0 00 014022'              MOVEM   T20BUF#                 ;  ...
005306' 263 17 0 00 000000               RTN                             ; return
```

```

                                        ; Set up next data pattern

005307' 350 01 0 00 014022'     TS20NE: AOS     1,T20BUF                ; point to next data pattern
005310' 307 01 0 00 000000#             CAIG    1,BUFF+5                ; done yet?
005311' 350 00 0 17 000000              AOS     (P)                     ; no - set up proper return
005312' 200 00 0 01 000000              MOVE    (1)                     ; get data pattern
005313' 201 02 0 00 005340'             MOVEI   2,TS20RS                ; get initial address of EBUF data
005314' 400 01 0 00 000000      TS20NO: SETZ    1,                      ; first shift doesn't include PLIN data
005315' 302 02 0 00 005340'             CAIE    2,TS20RS                ;   so check if first shift and if not
005316' 505 01 0 00 741777              HRLI    1,741777                ;   get initial PLIN data (left 8 bits)
005317' 246 00 0 00 000004              LSHC    4                       ; shift 4 bits
005320' 202 00 0 02 000000              MOVEM   (2)                     ; save it
005321' 350 00 0 00 000002              AOS     2                       ; point to next address
005322' 307 02 0 00 005351'             CAIG    2,TS20RS+11             ; done yet?
005323' 254 00 0 00 005314'             JRST    TS20NO                  ; no - loop till done
005324' 402 00 0 00 014045'             SETZM   TS20NM#                 ; initialize number of shifts

                                        ; Read the next word

005325' 201 01 0 00 000200              MOVEI   1,200                   ; get start address
005326' 242 01 0 00 000001              LSH     1,1                     ; position correctly
005327' 260 17 0 00 005210*             GO      LDRAR                   ; load the register
005330' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
005331' 260 17 0 00 005216*             GO      LDCSR                   ; start up the port
005332' 400 01 0 00 000000              SETZ    1,                      ; clear data
005333' 260 17 0 00 005331*             GO      LDCSR                   ; stop the port
005334' 263 17 0 00 000000              RTN                             ; return

                                        ; Increment shift count

005335' 201 00 0 00 000004      TS20NN: MOVEI   4                       ; get a 4
005336' 272 00 0 00 014045'             ADDM    TS20NM                  ; add it
005337' 263 17 0 00 000000              RTN                             ; return

005340'                         TS20RS: BLOCK   12                      ; correct EBUF data

                                        ; Error messages

005352' 240000  005354'         MA20:   CALL!TXNOT!MA20PN               ; print data pattern ...
005353' 000000000000#                   LAST!CALL!TXALL!TTPNT           ; print test data

005354' 037 00 0 00 013336'     MA20PN: TMSGC   <Data >                 ; get data and
005355' 200 01 0 00 014022'             MOVE    1,T20BUF                ;   print it
005356' 200 00 0 01 000000              MOVE    (1)
005357' 037 13 0 00 000000              PNTHW
005360' 037 00 0 00 013372'             TMSG    < shifted up from PLIN >
005361' 200 00 0 00 014045'             MOVE    TS20NM                  ; get shift count
005362' 037 15 0 00 000000              PNTDEC
005363' 037 00 0 00 013343'             TMSG    <. bits, result in EBUF>
005364' 263 17 0 00 000000              RTN
```

```

                                        ; Microcode - Shift 4 bits

005365' 010001  010000                  T20M:   MWORD   <ADDR=100,JMAP,J=101,D=1>                       ; 100
005366' 001000  000040
005367' 010100  000122                          MWORD   <CONT,D=1,SELF,MGC=122>                         ; 101
005370' 001000  003340
005371' 010200  000200                          MWORD   <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>              ; 102
005372' 732002  002340
005373' 010351  000000                          MWORD   <JMAP,J=5100,D=1>                               ; 103
005374' 001000  000040

                                        ;       Routine to read a word from CBUS into the formatter.

005375' 020002  010000                          MWORD   <ADDR=200,JMAP,J=201>                           ; 200
005376' 000000  000040
005377' 020102  030000                          MWORD   <CJP,J=203,D=1,CENA,CCCA>                       ; 201
005400' 001400  000060
005401' 020202  010000                          MWORD   <JMAP,J=201,D=1>                                ; 202
005402' 001000  000040
005403' 020300  000004                          MWORD   <CONT,D=1,SELC,MGC=4>                           ; 203
005404' 001000  004340
005405' 020400  000200                          MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>              ; 204
005406' 732001  402340
005407' 020500  000000                          MWORD   <JZ>                                            ; 205
005410' 000000  000000

                                        ;       Initialization

005411' 100010  010020                          MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>             ; 1000
005412' 000000  002040
005413' 100100  000200                          MWORD   <CONT,D=1,SELC,MGC=200>                         ; 1001
005414' 001000  004340
005415' 100200  310360                          MWORD   <LDCT,J=31,SDO,OR,D=2,SKCN,MGC=360>             ; 1002
005416' 732000  240300
005417' 100310  030000                          MWORD   <RPCT,J=1003,SOA,OR,D=5>                        ; 1003
005420' 435000  000220
005421' 100400  002040                          MWORD   <CONT,SOA,OR,D=1,OENA,SELM,MGC=40>              ; 1004
005422' 431000  002340
005423' 100500  000006                          MWORD   <JZ,SDO,D=1,SELP,MGC=6>                         ; 1005
005424' 701000  001000
005425' 000000  000000                          MWORD   <ADDR=0,JZ>                                     ; 0
005426' 000000  000000

                                        ;       Initialization step

005427' 500050  010100                          MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>            ; 5000
005430' 000000  004040
005431' 500100  000020                          MWORD   <JZ,SELM,MGC=20>                                ; 5001
005432' 000000  002000

                                        ; Subroutine - Passed - Write 1 to bit 12 of CSR register

005433' 510000  000000                          MWORD   <ADDR=5100,CONT,SDO,AND.B=2,D=2>                ; 5100
005434' 742001  000340
```

```
005435' 510100  000000                   MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                           ; 5101
005436' 302001  000740
005437' 510200  260000                   MWORD   <LDCT,J=26>                                           ; 5102
005440' 000000  000300
005441' 510351  030000                   MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>                    ; 5103
005442' 102021  000220
005443' 510400  002004                   MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>                 ; 5104
005444' 431020  005340
005445' 510551  070040                   MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>                ; 5105
005446' 001400  015060
005447' 510651  050000                   MWORD   <JMAP,J=5105,D=1>                                     ; 5106
005450' 001000  000040
005451' 510700  000200                   MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>                    ; 5107
005452' 431010  005340
005453' 511051  102004                   MWORD   <JMAP,J=5110,SOA,A=4,OR,D=1,OENA,SELE,MGC=4>          ; 5110
005454' 431040  005040
005455' 777777  777777                   -1
```

```

                                     ;#*****************************************************************
                                     ;* Test 21 - Fmtr Up Shift 8 (from PLIN) Test - SELRHTNIB=0
                                     ;
                                     ; Description:  Verify that the formatter can shift various  data
                                     ;               patterns up 8 bits at a time while inserting bits
                                     ;               from the PLIN into the formatter. The data loaded
                                     ;               into the PLIN initially is 017.
                                     ;
                                     ; Procedure:    KL> Port Clear
                                     ;               KL> Set up data transfer for data patterns
                                     ;
                                     ;               UC> Read a data pattern over the CBUS
                                     ;               UC> Load PLIN with 017
                                     ;
                                     ;               UC> Shift up 8 bits
                                     ;               UC> Read formatter contents and write into EBUF
                                     ;               KL> Read EBUF and verify the data
                                     ;               KL> Start up the ucode to shift again
                                     ;
                                     ;               Repeat the above shifting then verifying EBUF
                                     ;                 contents 5 times, until the data becomes
                                     ;                 all zero.
                                     ;
                                     ;               Do for data patterns:    0's
                                     ;                                        1's
                                     ;                                        525252,,525252
                                     ;                                        000000,,007777
                                     ;                                        000000,,007070
                                     ;                                        000000,,000707
                                     ;
                                     ; Failure:      ---
                                     ;#*****************************************************************

                                     ; Test data

005456' 254 00 0 00 005466'          TSTC21: JRST    TG21                    ; go start test
005457' 100604  000021                       CBUS!NDMP!NDCB!ZCBUS!21         ; test mask
005460' 005645' 013411'                      T21M,,[ASCIZ ^Fmtr Up Shift 8 (from PLIN) Test - SELRHTNIB=0^]
005461' 013263' 012716'                      [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
005462' 000000  005736'                      TSTC22                          ; failure test table
005463' 000000  006215'                      TSTC23                          ;   ...
005464' 000000  006473'                      TSTC24
005465' 777777  777777                       -1
```

```

                                        ; Start test

005466' 201 00 0 00 000000'     TG21:   MOVEI   Z9                              ; get address of module start
005467' 260 17 0 00 005156*             GO      TRACE                           ; handle trace output
005470' 201 01 0 00 005645'             MOVEI   1,T21M                          ; set up microcode address
005471' 260 17 0 00 005160*             GO      TLOAD                           ; load/verify it
005472' 263 17 0 00 000000              RTN                                     ; failed - exit test

                                        ; Initialization

005473' 400 15 0 00 000000      TL21:   SETZ    ERFLG,                          ; clear error flag
005474' 260 17 0 00 005163*             GO      IPACLR                          ; clear port
005475' 402 00 0 00 005164*             SETZM   TSTSUB                          ; initialize subtest number
                                        MOVSI   [000000,,000000                 ; get address of buffer
                                                 777777,,777777                 ;   contents
                                                 525252,,525252
                                                 000000,,007777
                                                 000000,,007070
005476' 205 00 0 00 013330'                      000000,,000707]
005477' 541 00 0 00 005201*             HRRI    BUFF                            ; build the BLT pointer
005500' 251 00 0 00 000000#             BLT     BUFF+5                          ; insert data into buffer

                                        ; First ensure the data transfer is reset

005501' 201 01 0 00 005000              MOVEI   1,5000                          ; get start address
005502' 242 01 0 00 000001              LSH     1,1                             ; position correctly
005503' 260 17 0 00 005327*             GO      LDRAR                           ; load the register
005504' 201 01 0 00 000010              MOVEI   1,MPRUN                         ; get MPRUN bit
005505' 260 17 0 00 005333*             GO      LDCSR                           ; start up the port
005506' 400 01 0 00 000000              SETZ    1,                              ; clear data
005507' 260 17 0 00 005505*             GO      LDCSR                           ; stop the port

                                        ; Now generate CCW list

005510' 200 01 0 00 013210'             MOVE    1,[CBUFSZ,,CBUF]                ; get pointer to control buffer
005511' 260 17 0 00 005200*             GO      CHINIT                          ; initialize software
005512' 551 01 0 00 005477*             HRRZI   1,BUFF                          ; buffer address
005513' 201 02 0 00 000006              MOVEI   2,6                             ; word count
005514' 400 03 0 00 000000              SETZ    3,                              ; clear zero fill specifier
005515' 260 17 0 00 005204*             GO      GENCCW                          ; generate a CCW list
005516' 201 06 0 00 005541'             MOVEI   6,TS21                          ; get test execute table address
```

```

                                      ; Start up the data transfer

005517' 201 01 0 00 001000                    MOVEI   1,1000                  ; get start address
005520' 242 01 0 00 000001                    LSH     1,1                     ; position correctly
005521' 260 17 0 00 005503*                   GO      LDRAR                   ; load the register
005522' 201 01 0 00 000010                    MOVEI   1,MPRUN                 ; get MPRUN bit
005523' 260 17 0 00 005507*                   GO      LDCSR                   ; start up the port
005524' 201 00 0 00 000001                    MOVEI   1                       ; set up delay of 1 ms
005525' 260 17 0 00 005214*                   GO      ODELAY                  ; wait a bit
005526' 400 01 0 00 000000                    SETZ    1,                      ; clear data
005527' 260 17 0 00 005523*                   GO      LDCSR                   ; stop the port

                                      ; Loop on test execute table entries

005530' 260 17 0 00 005217*           TA21:   GO      TEXEC                   ; execute table entry
005531' 254 00 0 00 005540'                   JRST    TX21                    ; end of table
005532' 254 00 0 00 005530'                   JRST    TA21                    ; keep looping after call
005533' 474 15 0 00 000000                    SETO    ERFLG,                  ; error - EBUF data incorrect

                                      ; Handle error printouts and scope looping

005534' 027 00 0 00 005632'                   SCOPER  MA21                    ; print error message
005535' 254 00 0 00 005473'                   JRST    TL21                    ; loop on error
005536' 254 00 0 00 005540'                   JRST    TX21                    ; altmode exit
005537' 322 15 0 00 005530'                   JUMPE   ERFLG,TA21              ; do next sstep table entry

                                      ; End of test

005540' 263 17 0 00 000000            TX21:   RTN                             ; return

                                      ; Test Execute Table, as:  (CMD,parameters)

005541' 140000  005571'               TS21:   TTABLE  (TCALL,TS21IN)          ; initialize stuff
005542' 200000  005574'               TS21L:  TTABLE  (TCALLC,TS21NE)         ; set up next data pattern

                                      ; Shifted 8 bits

0     3' 140000  005622'              TS21A:  TTABLE  (TCALL,TS21NN)          ; increment shift count
(   544' 400000  000100                       TTABLE  (TSSTAR,100)            ; start up port
    545' 240001  005625'                      TTABLE  (TCHECK,1,TS21RS)       ; check if completed
005546' 340000  000000                        TTABLE  (TEXIT)                 ; exit if error yet

                                      ; Shifted 16 bits

005547' 140000  005622'                       TTABLE  (TCALL,TS21NN)          ; increment shift count
005550' 400000  000100                        TTABLE  (TSSTAR,100)            ; start up port
005551' 240001  005626'                       TTABLE  (TCHECK,1,TS21RS+1)     ; check if completed
005552' 340000  000000                        TTABLE  (TEXIT)                 ; exit if error yet

```

```

                                          ; Shifted 24 bits

005553' 140000  005622'                          TTABLE  (TCALL,TS21NN)          ; increment shift count
005554' 400000  000100                           TTABLE  (TSSTAR,100)            ; start up port
005555' 240001  005627'                          TTABLE  (TCHECK,1,TS21RS+2)     ; check if completed
005556' 340000  000000                           TTABLE  (TEXIT)                 ; exit if error yet

                                          ; Shifted 32 bits

005557' 140000  005622'                          TTABLE  (TCALL,TS21NN)          ; increment shift count
005560' 400000  000100                           TTABLE  (TSSTAR,100)            ; start up port
005561' 240001  005630'                          TTABLE  (TCHECK,1,TS21RS+3)     ; check if completed
005562' 340000  000000                           TTABLE  (TEXIT)                 ; exit if error yet

                                          ; Shifted 40 bits

005563' 140000  005622'                          TTABLE  (TCALL,TS21NN)          ; increment shift count
005564' 400000  000100                           TTABLE  (TSSTAR,100)            ; start up port
005565' 240001  005631'                          TTABLE  (TCHECK,1,TS21RS+4)     ; check if completed
005566' 340000  000000                           TTABLE  (TEXIT)                 ; exit if error yet
005567' 300000  005542'                          TTABLE  (TJRST,TS21L)           ; loop till done
005570' 000000  000000                           TTABLE  (TLAST)

                                          ; Initialize stuff

005571' 201 00 0 00 000000#               TS21IN: MOVEI   BUFF-1                 ; initialize buffer pointer
005572' 202 00 0 00 014023'                       MOVEM   T21BUF#                ;  ...
005573' 263 17 0 00 000000                        RTN                            ; return

                                          ; Set up next data pattern

005574' 350 01 0 00 014023'               TS21NE: AOS     1,T21BUF               ; point to next data pattern
005575' 307 01 0 00 000000#                       CAIG    1,BUFF+5               ; done yet?
005576' 350 00 0 17 000000                        AOS     (P)                    ; no - set up proper return
005577' 200 00 0 01 000000                        MOVE    (1)                    ; get data pattern
005600' 201 02 0 00 005625'                       MOVEI   2,TS21RS               ; get initial address of EBUF data
005601' 400 01 0 00 000000                TS21NO: SETZ    1,                     ; first shift doesn't include PLIN data
005602' 302 02 0 00 005625'                       CAIE    2,TS21RS               ;   so check if first shift and if not
005603' 505 01 0 00 741777                        HRLI    1,741777               ;   get initial PLIN data (left 8 bits)
005604' 246 00 0 00 000010                        LSHC    8                      ; shift 8 bits
005605' 202 00 0 02 000000                        MOVEM   (2)                    ; save it
005606' 350 00 0 00 000002                        AOS     2                      ; point to next address
005607' 307 02 0 00 005631'                       CAIG    2,TS21RS+4             ; done yet?
005610' 254 00 0 00 005601'                       JRST    TS21NO                 ; no - loop till done
005611' 402 00 0 00 014046'                       SETZM   TS21NM#                ; initialize number of shifts
```

```

                                  ; Read the next word

005612' 201 01 0 00 000200                MOVEI   1,200                  ; get start address
005613' 242 01 0 00 000001                LSH     1,1                    ; position correctly
005614' 260 17 0 00 005521*               GO      LDRAR                  ; load the register
005615' 201 01 0 00 000010                MOVEI   1,MPRUN                ; get MPRUN bit
005616' 260 17 0 00 005527*               GO      LDCSR                  ; start up the port
005617' 400 01 0 00 000000                SETZ    1,                     ; clear data
005620' 260 17 0 00 005616*               GO      LDCSR                  ; stop the port
005621' 263 17 0 00 000000                RTN                            ; return

                                  ; Increment shift count

005622' 201 00 0 00 000010        TS21NN: MOVEI   8                      ; get an 8
005623' 272 00 0 00 014046'               ADDM    TS21NM                 ; add it
005624' 263 17 0 00 000000                RTN                            ; return

005625'                           TS21RS: BLOCK   5                      ; correct EBUF data

                                  ; Error messages

005632' 240000  005634'           MA21:   CALL!TXNOT!MA21PN              ; print data pattern ...
005633' 000000000000#                     LAST!CALL!TXALL!TTPNT          ; print test data

005634' 037 00 0 00 013336'       MA21PN: TMSGC   <Data >                ; get data and
005635' 200 01 0 00 014023'               MOVE    1,T21BUF               ;   print it
005636' 200 00 0 01 000000                MOVE    (1)
005637' 037 13 0 00 000000                PNTHW
005640' 037 00 0 00 013372'               TMSG    < shifted up from PLIN >
005641' 200 00 0 00 014046'               MOVE    TS21NM                 ; get shift count
005642' 037 15 0 00 000000                PNTDEC
005643' 037 00 0 00 013343'               TMSG    <. bits, result in EBUF>
005644' 263 17 0 00 000000                RTN
```

```

                                         ; Microcode - Shift 8 bits

005645' 010001  010000                   T21M:   MWORD   <ADDR=100,JMAP,J=101,D=1>                  ; 100
005646' 001000  000040
005647' 010100  000220                           MWORD   <CONT,D=1,SELF,MGC=220>                    ; 101
005650' 001000  003340
005651' 010200  000200                           MWORD   <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>         ; 102
005652' 732002  002340
005653' 010351  000000                           MWORD   <JMAP,J=5100,D=1>                          ; 103
005654' 001000  000040

                                         ;       Routine to read a word from CBUS into the formatter.

005655' 020002  010000                           MWORD   <ADDR=200,JMAP,J=201>                      ; 200
005656' 000000  000040
005657' 020102  030000                           MWORD   <CJP,J=203,D=1,CENA,CCCA>                  ; 201
005660' 001400  000060
005661' 020202  010000                           MWORD   <JMAP,J=201,D=1>                           ; 202
005662' 001000  000040
005663' 020300  000004                           MWORD   <CONT,D=1,SELC,MGC=4>                      ; 203
005664' 001000  004340
005665' 020400  000200                           MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>         ; 204
005666' 732001  402340
005667' 020500  000000                           MWORD   <JZ>                                       ; 205
005670' 000000  000000

                                         ;       Initialization

005671' 100010  010020                           MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>        ; 1000
005672' 000000  002040
005673' 100100  000200                           MWORD   <CONT,D=1,SELC,MGC=200>                    ; 1001
005674' 001000  004340
005675' 100200  310017                           MWORD   <LDCT,J=31,SDO,OR,D=2,SKCN,MGC=017>        ; 1002
005676' 732000  240300
005677' 100310  030000                           MWORD   <RPCT,J=1003,SOA,OR,D=5>                   ; 1003
005700' 435000  000220
005701' 100400  002040                           MWORD   <CONT,SOA,OR,D=1,OENA,SELM,MGC=40>         ; 1004
005702' 431000  002340
005703' 100500  000006                           MWORD   <JZ,SDO,D=1,SELP,MGC=6>                    ; 1005
005704' 701000  001000
005705' 000000  000000                           MWORD   <ADDR=0,JZ>                                ; 0
005706' 000000  000000

                                         ;       Initialization step

005707' 500050  010100                           MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>       ; 5000
005710' 000000  004040
005711' 500100  000020                           MWORD   <JZ,SELM,MGC=20>                           ; 5001
005712' 000000  002000

                                         ; Subroutine - Passed - Write 1 to bit 12 of CSR register

005713' 510000  000000                           MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>           ; 5100
005714' 742001  000340
```

```
4792 005715' 510100  000000          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                   ; 5101
4793 005716' 302001  000740
4794 005717' 510200  260000          MWORD   <LDCT,J=26>                                   ; 5102
4795 005720' 000000  000300
4796 005721' 510351  030000          MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>            ; 5103
4797 005722' 102021  000220
4798 005723' 510400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>         ; 5104
4799 005724' 431020  005340
4800 005725' 510551  070040          MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>        ; 5105
4801 005726' 001400  015060
4802 005727' 510651  050000          MWORD   <JMAP,J=5105,D=1>                             ; 5106
4803 005730' 001000  000040
4804 005731' 510700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>            ; 5107
4805 005732' 431010  005340
4806 005733' 511051  102004          MWORD   <JMAP,J=5110,SOA,A=4,OR,D=1,OENA,SELE,MGC=4>  ; 5110
4807 005734' 431040  005040
4808 005735' 777777  777777          -1
```

```
                                        ;#**********************************************************************
                                        ;* Test 22 - Fmtr Up Shift 8 (from PLIN) Test - SELRHTNIB=0
                                        ;
                                        ; Description:  Verify that the formatter can shift various  data
                                        ;               patterns up 8 bits at a time while inserting bits
                                        ;               from the PLIN into the formatter. The data loaded
                                        ;               into the PLIN initially is 360.
                                        ;
                                        ; Procedure:    KL> Port Clear
                                        ;               KL> Set up data transfer for data patterns
                                        ;
                                        ;               UC> Read a data pattern over the CBUS
                                        ;               UC> Load PLIN with 360
                                        ;
                                        ;               UC> Shift up 8 bits
                                        ;               UC> Read formatter contents and write into EBUF
                                        ;               KL> Read EBUF and verify the data
                                        ;               KL> Start up the ucode to shift again
                                        ;
                                        ;               Repeat the above shifting then verifying EBUF
                                        ;                 contents 5 times, until the data becomes
                                        ;                 all zero.
                                        ;
                                        ;               Do for data patterns:   0's
                                        ;                                       1's
                                        ;                                       525252,,525252
                                        ;                                       000000,,007777
                                        ;                                       000000,,007070
                                        ;                                       000000,,000707
                                        ;
                                        ; Failure:      ---
                                        ;#**********************************************************************

                                        ; Test data

 005736'  254 00 0 00 005747'           TSTC22: JRST    TG22                    ; go start test
 005737'  100604  000022                        CBUS!NDMP!NDCB!ZCBUS!22         ; test mask
 005740'  006124' 013411'                       T22M,,[ASCIZ ^Fmtr Up Shift 8 (from PLIN) Test - SELRHTNIB=0^]
 005741'  013263' 012716'                       [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
 005742'  000000  006215'                       TSTC23                          ; failure test table
 005743'  000000  006473'                       TSTC24                          ;   ...
 005744'  000000  006754'                       TSTC25
 005745'  000000  007062'                       TSTC26
 005746'  777777  777777                        -1
```

```

                                        ; Start test

005747' 201 00 0 00 000000'     TG22:   MOVEI   Z9                      ; get address of module start
005750' 260 17 0 00 005467*             GO      TRACE                   ; handle trace output
005751' 201 01 0 00 006124'             MOVEI   1,T22M                  ; set up microcode address
005752' 260 17 0 00 005471*             GO      TLOAD                   ; load/verify it
005753' 263 17 0 00 000000              RTN                             ; failed - exit test

                                        ; Initialization

005754' 400 15 0 00 000000      TL22:   SETZ    ERFLG,                  ; clear error flag
005755' 260 17 0 00 005474*             GO      IPACLR                  ; clear port
005756' 402 00 0 00 005475*             SETZM   TSTSUB                  ; initialize subtest number
                                        MOVSI   [000000,,000000         ; get address of buffer
                                                 777777,,777777         ;   contents
                                                 525252,,525252
                                                 000000,,007777
                                                 000000,,007070
005757' 205 00 0 00 013330'                      000000,,000707]
005760' 541 00 0 00 005512*             HRRI    BUFF                    ; build the BLT pointer
005761' 251 00 0 00 000000#             BLT     BUFF+5                  ; insert data into buffer

                                        ; First ensure the data transfer is reset

005762' 201 01 0 00 005000              MOVEI   1,5000                  ; get start address
005763' 242 01 0 00 000001              LSH     1,1                     ; position correctly
005764' 260 17 0 00 005614*             GO      LDRAR                   ; load the register
005765' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
005766' 260 17 0 00 005620*             GO      LDCSR                   ; start up the port
005767' 400 01 0 00 000000              SETZ    1,                      ; clear data
005770' 260 17 0 00 005766*             GO      LDCSR                   ; stop the port

                                        ; Now generate CCW list

005771' 200 01 0 00 013210'             MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
005772' 260 17 0 00 005511*             GO      CHINIT                  ; initialize software
005773' 551 01 0 00 005760*             HRRZI   1,BUFF                  ; buffer address
005774' 201 02 0 00 000006              MOVEI   2,6                     ; word count
005775' 400 03 0 00 000000              SETZ    3,                      ; clear zero fill specifier
005776' 260 17 0 00 005515*             GO      GENCCW                  ; generate a CCW list
005777' 201 06 0 00 006022'             MOVEI   6,TS22                  ; get test execute table address

                                        ; Start up the data transfer

006000' 201 01 0 00 001000              MOVEI   1,1000                  ; get start address
006001' 242 01 0 00 000001              LSH     1,1                     ; position correctly
006002' 260 17 0 00 005764*             GO      LDRAR                   ; load the register
006003' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
006004' 260 17 0 00 005770*             GO      LDCSR                   ; start up the port
006005' 201 00 0 00 000001              MOVEI   1                       ; set up delay of 1 ms
006006' 260 17 0 00 005525*             GO      ODELAY                  ; wait a bit
006007' 400 01 0 00 000000              SETZ    1,                      ; clear data
006010' 260 17 0 00 006004*             GO      LDCSR                   ; stop the port
```

```

                                        ; Loop on test execute table entries

006011' 260 17 0 00 005530*     TA22:   GO      TEXEC                   ; execute table entry
006012' 254 00 0 00 006021'             JRST    TX22                    ; end of table
006013' 254 00 0 00 006011'             JRST    TA22                    ; keep looping after call
006014' 474 15 0 00 000000              SETO    ERFLG,                  ; error - EBUF data incorrect

                                        ; Handle error printouts and scope looping

006015' 027 00 0 00 006111'             SCOPER  MA22                    ; print error message
006016' 254 00 0 00 005754'             JRST    TL22                    ; loop on error
006017' 254 00 0 00 006021'             JRST    TX22                    ; altmode exit
006020' 322 15 0 00 006011'             JUMPE   ERFLG,TA22              ; do next sstep table entry

                                        ; End of test

006021' 263 17 0 00 000000      TX22:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

006022' 140000  006052'         TS22:   TTABLE  (TCALL,TS22IN)          ; initialize stuff
006023' 200000  006055'         TS22L:  TTABLE  (TCALLC,TS22NE)         ; set up next data pattern

                                        ; Shifted 8 bits

006024' 140000  006101'         TS22A:  TTABLE  (TCALL,TS22NN)          ; increment shift count
006025' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
006026' 240001  006104'                 TTABLE  (TCHECK,1,TS22RS)       ; check if completed
006027' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 16 bits

006030' 140000  006101'                 TTABLE  (TCALL,TS22NN)          ; increment shift count
006031' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
006032' 240001  006105'                 TTABLE  (TCHECK,1,TS22RS+1)     ; check if completed
006033' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 24 bits

006034' 140000  006101'                 TTABLE  (TCALL,TS22NN)          ; increment shift count
006035' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
006036' 240001  006106'                 TTABLE  (TCHECK,1,TS22RS+2)     ; check if completed
006037' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

```

```

                                        ; Shifted 32 bits

006040' 140000  006101'                         TTABLE  (TCALL,TS22NN)          ; increment shift count
006041' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
006042' 240001  006107'                         TTABLE  (TCHECK,1,TS22RS+3)     ; check if completed
006043' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 40 bits

006044' 140000  006101'                         TTABLE  (TCALL,TS22NN)          ; increment shift count
006045' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
006046' 240001  006110'                         TTABLE  (TCHECK,1,TS22RS+4)     ; check if completed
006047' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet
006050' 300000  006023'                         TTABLE  (TJRST,TS22L)           ; loop till done
006051' 000000  000000                          TTABLE  (TLAST)

                                        ; Initialize stuff

006052' 201 00 0 00 000000#             TS22IN: MOVEI   BUFF-1                  ; initialize buffer pointer
006053' 202 00 0 00 014024'                     MOVEM   T22BUF#                 ;  ...
006054' 263 17 0 00 000000                      RTN                             ; return

                                        ; Set up next data pattern

006055' 350 01 0 00 014024'             TS22NE: AOS     1,T22BUF                ; point to next data pattern
006056' 307 01 0 00 000000#                     CAIG    1,BUFF+5                ; done yet?
006057' 350 00 0 17 000000                      AOS     (P)                     ; no - set up proper return
006060' 200 00 0 01 000000                      MOVE    (1)                     ; get data pattern
006061' 201 02 0 00 006104'                     MOVEI   2,TS22RS                ; get initial address of EBUF data
006062' 505 01 0 00 037777              TS22NO: HRLI    1,37777                 ; get initial PLIN data (left 8 bits)
006063' 246 00 0 00 000010                      LSHC    8                       ; shift 8 bits
006064' 202 00 0 02 000000                      MOVEM   (2)                     ; save it
006065' 350 00 0 00 000002                      AOS     2                       ; point to next address
006066' 307 02 0 00 006110'                     CAIG    2,TS22RS+4              ; done yet?
006067' 254 00 0 00 006062'                     JRST    TS22NO                  ; no - loop till done
006070' 402 00 0 00 014047'                     SETZM   TS22NM#                 ; initialize number of shifts

                                        ; Read the next word

006071' 201 01 0 00 000200                      MOVEI   1,200                   ; get start address
006072' 242 01 0 00 000001                      LSH     1,1                     ; position correctly
006073' 260 17 0 00 006002*                     GO      LDRAR                   ; load the register
006074' 201 01 0 00 000010                      MOVEI   1,MPRUN                 ; get MPRUN bit
006075' 260 17 0 00 006010*                     GO      LDCSR                   ; start up the port
006076' 400 01 0 00 000000                      SETZ    1,                      ; clear data
006077' 260 17 0 00 006075*                     GO      LDCSR                   ; stop the port
006100' 263 17 0 00 000000                      RTN                             ; return
```

```

                                        ; Increment shift count

006101' 201 00 0 00 000010      TS22NN: MOVEI   8                       ; get a 8
006102' 272 00 0 00 014047'             ADDM    TS22NM                  ; add it
006103' 263 17 0 00 000000              RTN                             ; return

006104'                         TS22RS: BLOCK   5                       ; correct EBUF data

                                ; Error messages

006111' 240000  006113'         MA22:   CALL!TXNOT!MA22PN               ; print data pattern ...
006112' 000000000000#                   LAST!CALL!TXALL!TTPNT           ; print test data

006113' 037 00 0 00 013336'     MA22PN: TMSGC   <Data >                 ; get data and
006114' 200 01 0 00 014024'             MOVE    1,T22BUF                ;   print it
006115' 200 00 0 01 000000              MOVE    (1)
006116' 037 13 0 00 000000              PNTHW
006117' 037 00 0 00 013372'             TMSG    < shifted up from PLIN >
006120' 200 00 0 00 014047'             MOVE    TS22NM                  ; get shift count
006121' 037 15 0 00 000000              PNTDEC
006122' 037 00 0 00 013343'             TMSG    <. bits, result in EBUF>
006123' 263 17 0 00 000000              RTN
```

```


                                  ; Microcode - Shift 8 bits

006124' 010001  010000            T22M:  MWORD   <ADDR=100,JMAP,J=101,D=1>                        ; 100
006125' 001000  000040
006126' 010100  000220                   MWORD   <CONT,D=1,SELF,MGC=220>                          ; 101
006127' 001000  003340
006130' 010200  000200                   MWORD   <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>               ; 102
006131' 732002  002340
006132' 010351  000000                   MWORD   <JMAP,J=5100,D=1>                                ; 103
006133' 001000  000040

                                  ;      Routine to read a word from CBUS into the formatter.

006134' 020002  010000                   MWORD   <ADDR=200,JMAP,J=201>                            ; 200
006135' 000000  000040
006136' 020102  030000                   MWORD   <CJP,J=203,D=1,CENA,CCCA>                        ; 201
006137' 001400  000060
006140' 020202  010000                   MWORD   <JMAP,J=201,D=1>                                 ; 202
006141' 001000  000040
006142' 020300  000004                   MWORD   <CONT,D=1,SELC,MGC=4>                            ; 203
006143' 001000  004340
006144' 020400  000200                   MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>               ; 204
006145' 732001  402340
006146' 020500  000000                   MWORD   <JZ>                                             ; 205
006147' 000000  000000

                                  ;      Initialization

006150' 100010  010020                   MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>              ; 1000
006151' 000000  002040
006152' 100100  000200                   MWORD   <CONT,D=1,SELC,MGC=200>                          ; 1001
006153' 001000  004340
006154' 100200  310360                   MWORD   <LDCT,J=31,SDO,OR,D=2,SKCN,MGC=360>              ; 1002
006155' 732000  240300
006156' 100310  030000                   MWORD   <RPCT,J=1003,SOA,OR,D=5>                         ; 1003
006157' 435000  000220
006160' 100400  002040                   MWORD   <CONT,SOA,OR,D=1,OENA,SELM,MGC=40>               ; 1004
006161' 431000  002340
006162' 100500  000006                   MWORD   <JZ,SDO,D=1,SELP,MGC=6>                          ; 1005
006163' 701000  001000
006164' 000000  000000                   MWORD   <ADDR=0,JZ>                                      ; 0
006165' 000000  000000

                                  ;      Initialization step

006166' 500050  010100                   MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>             ; 5000
006167' 000000  004040
006170' 500100  000020                   MWORD   <JZ,SELM,MGC=20>                                 ; 5001
006171' 000000  002000

                                  ; Subroutine - Passed - Write 1 to bit 12 of CSR register

006172' 510000  000000                   MWORD   <ADDR=5100,CONT,SDO,AND.B=2,D=2>                 ; 5100
006173' 742001  000340
```

```
 5079  006174' 510100  000000                    MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                          ; 5101
 5080  006175' 302001  000740
 5081  006176' 510200  260000                    MWORD   <LDCT,J=26>                                          ; 5102
 5082  006177' 000000  000300
 5083  006200' 510351  030000                    MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>                   ; 5103
 5084  006201' 102021  000220
 5085  006202' 510400  002004                    MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>                ; 5104
 5086  006203' 431020  005340
 5087  006204' 510551  070040                    MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>               ; 5105
 5088  006205' 001400  015060
 5089  006206' 510651  050000                    MWORD   <JMAP,J=5105,D=1>                                    ; 5106
 5090  006207' 001000  000040
 5091  006210' 510700  000200                    MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>                   ; 5107
 5092  006211' 431010  005340
 5093  006212' 511051  102004                    MWORD   <JMAP,J=5110,SOA,A=4,OR,D=1,OENA,SELE,MGC=4>         ; 5110
 5094  006213' 431040  005040
 5095  006214' 777777  777777                    -1
```

```

                                ;#**********************************************************************
                                ;* Test 23 - Fmtr Up Shift 8 (from PLIN) Test - SELRHTNIB=1
                                ;
                                ; Description:  Verify that the formatter can shift various  data
                                ;               patterns up 8 bits at a time while inserting bits
                                ;               from the PLIN into the formatter. The data loaded
                                ;               into the PLIN initially is 017.
                                ;
                                ; Procedure:    KL> Port Clear
                                ;               KL> Set up data transfer for data patterns
                                ;
                                ;               UC> Read a data pattern over the CBUS
                                ;               UC> Load PLIN with 017
                                ;
                                ;               UC> Shift up 8 bits
                                ;               UC> Read formatter contents and write into EBUF
                                ;               KL> Read EBUF and verify the data
                                ;               KL> Start up the ucode to shift again
                                ;
                                ;               Repeat the above shifting then verifying EBUF
                                ;                 contents 5 times, until the data becomes
                                ;                 all zero.
                                ;
                                ;               Do for data patterns:   0's
                                ;                                       1's
                                ;                                       525252,,525252
                                ;                                       000000,,007777
                                ;                                       000000,,007070
                                ;                                       000000,,000707
                                ;
                                ; Failure:      ---
                                ;#**********************************************************************

                                ; Test data

006215' 254 00 0 00 006225'     TSTC23: JRST    TG23                    ; go start test
006216' 100604  000023                  CBUS!NDMP!NDCB!ZCBUS!23         ; test mask
006217' 006402' 013423'                 T23M,,[ASCIZ ^Fmtr Up Shift 8 (from PLIN) Test - SELRHTNIB=1^]
006220' 013263' 012716'                 [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
006221' 000000  006473'                 TSTC24                          ; failure test table
006222' 000000  006754'                 TSTC25                          ;   ...
006223' 000000  007062'                 TSTC26
006224' 777777  777777                  -1

                                ; Start test

006225' 201 00 0 00 000000'     TG23:   MOVEI   Z9                      ; get address of module start
006226' 260 17 0 00 005750*             GO      TRACE                   ; handle trace output
006227' 201 01 0 00 006402'             MOVEI   1,T23M                  ; set up microcode address
006230' 260 17 0 00 005752*             GO      TLOAD                   ; load/verify it
006231' 263 17 0 00 000000              RTN                             ; failed - exit test
```

```

                                        ; Initialization

006232' 400 15 0 00 000000      TL23:   SETZ    ERFLG,                  ; clear error flag
006233' 260 17 0 00 005755*             GO      IPACLR                  ; clear port
006234' 402 00 0 00 005756*             SETZM   TSTSUB                  ; initialize subtest number
                                        MOVSI   [000000,,000000         ; get address of buffer
                                                 777777,,777777         ;   contents
                                                 525252,,525252
                                                 000000,,007777
                                                 000000,,007070
006235' 205 00 0 00 013330'                      000000,,000707]
006236' 541 00 0 00 005773*             HRRI    BUFF                    ; build the BLT pointer
006237' 251 00 0 00 000000#             BLT     BUFF+5                  ; insert data into buffer

                                ; First ensure the data transfer is reset

006240' 201 01 0 00 005000              MOVEI   1,5000                  ; get start address
006241' 242 01 0 00 000001              LSH     1,1                     ; position correctly
006242' 260 17 0 00 006073*             GO      LDRAR                   ; load the register
006243' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
006244' 260 17 0 00 006077*             GO      LDCSR                   ; start up the port
006245' 400 01 0 00 000000              SETZ    1,                      ; clear data
006246' 260 17 0 00 006244*             GO      LDCSR                   ; stop the port

                                ; Now generate CCW list

006247' 200 01 0 00 013210'             MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
006250' 260 17 0 00 005772*             GO      CHINIT                  ; initialize software
006251' 551 01 0 00 006236*             HRRZI   1,BUFF                  ; buffer address
006252' 201 02 0 00 000006              MOVEI   2,6                     ; word count
006253' 400 03 0 00 000000              SETZ    3,                      ; clear zero fill specifier
006254' 260 17 0 00 005776*             GO      GENCCW                  ; generate a CCW list
006255' 201 06 0 00 006300'             MOVEI   6,TS23                  ; get test execute table address

                                ; Start up the data transfer

006256' 201 01 0 00 001000              MOVEI   1,1000                  ; get start address
006257' 242 01 0 00 000001              LSH     1,1                     ; position correctly
006260' 260 17 0 00 006242*             GO      LDRAR                   ; load the register
006261' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
006262' 260 17 0 00 006246*             GO      LDCSR                   ; start up the port
006263' 201 00 0 00 000001              MOVEI   1                       ; set up delay of 1 ms
006264' 260 17 0 00 006006*             GO      ODELAY                  ; wait a bit
006265' 400 01 0 00 000000              SETZ    1,                      ; clear data
006266' 260 17 0 00 006262*             GO      LDCSR                   ; stop the port
```

```

                                        ; Loop on test execute table entries

006267' 260 17 0 00 006011*     TA23:   GO      TEXEC                   ; execute table entry
006270' 254 00 0 00 006277'             JRST    TX23                    ; end of table
006271' 254 00 0 00 006267'             JRST    TA23                    ; keep looping after call
006272' 474 15 0 00 000000              SETO    ERFLG,                  ; error - EBUF data incorrect

                                        ; Handle error printouts and scope looping

006273' 027 00 0 00 006367'             SCOPER  MA23                    ; print error message
006274' 254 00 0 00 006232'             JRST    TL23                    ; loop on error
006275' 254 00 0 00 006277'             JRST    TX23                    ; altmode exit
006276' 322 15 0 00 006267'             JUMPE   ERFLG,TA23              ; do next sstep table entry

                                        ; End of test

006277' 263 17 0 00 000000      TX23:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

006300' 140000  006330'         TS23:   TTABLE  (TCALL,TS23IN)          ; initialize stuff
006301' 200000  006333'         TS23L:  TTABLE  (TCALLC,TS23NE)         ; set up next data pattern

                                        ; Shifted 8 bits

006302' 140000  006357'         TS23A:  TTABLE  (TCALL,TS23NN)          ; increment shift count
006303' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
006304' 240001  006362'                 TTABLE  (TCHECK,1,TS23RS)       ; check if completed
006305' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 16 bits

006306' 140000  006357'                 TTABLE  (TCALL,TS23NN)          ; increment shift count
006307' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
006310' 240001  006363'                 TTABLE  (TCHECK,1,TS23RS+1)     ; check if completed
006311' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 24 bits

006312' 140000  006357'                 TTABLE  (TCALL,TS23NN)          ; increment shift count
006313' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
006314' 240001  006364'                 TTABLE  (TCHECK,1,TS23RS+2)     ; check if completed
006315' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 32 bits

006316' 140000  006357'                 TTABLE  (TCALL,TS23NN)          ; increment shift count
006317' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
006320' 240001  006365'                 TTABLE  (TCHECK,1,TS23RS+3)     ; check if completed
006321' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

```

```

                                ; Shifted 40 bits

006322' 140000  006357'                 TTABLE  (TCALL,TS23NN)          ; increment shift count
006323' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
006324' 240001  006366'                 TTABLE  (TCHECK,1,TS23RS+4)     ; check if completed
006325' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet
006326' 300000  006301'                 TTABLE  (TJRST,TS23L)           ; loop till done
006327' 000000  000000                  TTABLE  (TLAST)

                                ; Initialize stuff

006330' 201 00 0 00 000000#     TS23IN: MOVEI   BUFF-1                  ; initialize buffer pointer
006331' 202 00 0 00 014025'             MOVEM   T23BUF#                 ; ...
006332' 263 17 0 00 000000              RTN                             ; return

                                ; Set up next data pattern

006333' 350 01 0 00 014025'     TS23NE: AOS     1,T23BUF                ; point to next data pattern
006334' 307 01 0 00 000000#             CAIG    1,BUFF+5                ; done yet?
006335' 350 00 0 17 000000              AOS     (P)                     ; no - set up proper return
006336' 200 00 0 01 000000              MOVE    (1)                     ; get data pattern
006337' 201 02 0 00 006362'             MOVEI   2,TS23RS                ; get initial address of EBUF data
006340' 505 01 0 00 037777      TS23NO: HRLI    1,37777                 ; get initial PLIN data (left 8 bits)
006341' 246 00 0 00 000010              LSHC    8                       ; shift 8 bits
006342' 202 00 0 02 000000              MOVEM   (2)                     ; save it
006343' 350 00 0 00 000002              AOS     2                       ; point to next address
006344' 307 02 0 00 006366'             CAIG    2,TS23RS+4              ; done yet?
006345' 254 00 0 00 006340'             JRST    TS23NO                  ; no - loop till done
006346' 402 00 0 00 014050'             SETZM   TS23NM#                 ; initialize number of shifts

                                ; Read the next word

006347' 201 01 0 00 000200              MOVEI   1,200                   ; get start address
006350' 242 01 0 00 000001              LSH     1,1                     ; position correctly
006351' 260 17 0 00 006260*             GO      LDRAR                   ; load the register
006352' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
006353' 260 17 0 00 006266*             GO      LDCSR                   ; start up the port
006354' 400 01 0 00 000000              SETZ    1,                      ; clear data
006355' 260 17 0 00 006353*             GO      LDCSR                   ; stop the port
006356' 263 17 0 00 000000              RTN                             ; return

                                ; Increment shift count

006357' 201 00 0 00 000010      TS23NN: MOVEI   8                       ; get a 8
006360' 272 00 0 00 014050'             ADDM    TS23NM                  ; add it
006361' 263 17 0 00 000000              RTN                             ; return

006362'                         TS23RS: BLOCK   5                       ; correct EBUF data
```

```

                                        ; Error messages

006367' 240000  006371'                 MA23:   CALL!TXNOT!MA23PN               ; print data pattern ...
006370' 000000000000#                           LAST!CALL!TXALL!TTPNT           ; print test data

006371' 037 00 0 00 013336'             MA23PN: TMSGC   <Data >                 ; get data and
006372' 200 01 0 00 014025'                     MOVE    1,T23BUF                ;   print it
006373' 200 00 0 01 000000                      MOVE    (1)
006374' 037 13 0 00 000000                      PNTHW
006375' 037 00 0 00 013372'                     TMSG    < shifted up from PLIN >
006376' 200 00 0 00 014050'                     MOVE    TS23NM                  ; get shift count
006377' 037 15 0 00 000000                      PNTDEC
006400' 037 00 0 00 013343'                     TMSG    <. bits, result in EBUF>
006401' 263 17 0 00 000000                      RTN

                                        ; Microcode - Shift 8 bits

006402' 010001  010000                  T23M:   MWORD   <ADDR=100,JMAP,J=101,D=1>                       ; 100
006403' 001000  000040
006404' 010100  000222                          MWORD   <CONT,D=1,SELF,MGC=222>                         ; 101
006405' 001000  003340
006406' 010200  000200                          MWORD   <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>              ; 102
006407' 732002  002340
006410' 010351  000000                          MWORD   <JMAP,J=5100,D=1>                               ; 103
006411' 001000  000040

                                        ;       Routine to read a word from CBUS into the formatter.

006412' 020002  010000                          MWORD   <ADDR=200,JMAP,J=201>                           ; 200
006413' 001000  000040
006414' 020102  030000                          MWORD   <CJP,J=203,D=1,CENA,CCCA>                       ; 201
006415' 001400  000060
006416' 020202  010000                          MWORD   <JMAP,J=201,D=1>                                ; 202
006417' 001000  000040
006420' 020300  000004                          MWORD   <CONT,D=1,SELC,MGC=4>                           ; 203
006421' 001000  004340
006422' 020400  000200                          MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>              ; 204
006423' 732001  402340
006424' 020500  000000                          MWORD   <JZ>                                            ; 205
006425' 000000  000000

                                        ;       Initialization

006426' 100010  010020                          MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>             ; 1000
006427' 000000  002040
006430' 100100  000200                          MWORD   <CONT,D=1,SELC,MGC=200>                         ; 1001
006431' 001000  004340
006432' 100200  310017                          MWORD   <LDCT,J=31,SDO,OR,D=2,SKCN,MGC=017>             ; 1002
006433' 732000  240300
006434' 100310  030000                          MWORD   <RPCT,J=1003,SOA,OR,D=5>                        ; 1003
006435' 435000  000220
006436' 100400  002040                          MWORD   <CONT,SOA,OR,D=1,OENA,SELM,MGC=40>              ; 1004
006437' 431000  002340
006440' 100500  000006                          MWORD   <JZ,SDO,D=1,SELP,MGC=6>                         ; 1005
```

```
 5350  006441' 701000  001000
 5351  006442' 000000  000000                      MWORD   <ADDR=0,JZ>                                   ; 0
 5352  006443' 000000  000000
 5353
 5354                                    ;         Initialization step
 5355
 5356  006444' 500050  010100                      MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>          ; 5000
 5357  006445' 000000  004040
 5358  006446' 500100  000020                      MWORD   <JZ,SELM,MGC=20>                              ; 5001
 5359  006447' 000000  002000
 5360
 5361                                    ; Subroutine - Passed - Write 1 to bit 12 of CSR register
 5362
 5363  006450' 510000  000000                      MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>              ; 5100
 5364  006451' 742001  000340
 5365  006452' 510100  000000                      MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                   ; 5101
 5366  006453' 302001  000740
 5367  006454' 510200  260000                      MWORD   <LDCT,J=26>                                   ; 5102
 5368  006455' 000000  000300
 5369  006456' 510351  030000                      MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>            ; 5103
 5370  006457' 102021  000220
 5371  006460' 510400  002004                      MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>         ; 5104
 5372  006461' 431020  005340
 5373  006462' 510551  070040                      MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>        ; 5105
 5374  006463' 001400  015060
 5375  006464' 510651  050000                      MWORD   <JMAP,J=5105,D=1>                             ; 5106
 5376  006465' 001000  000040
 5377  006466' 510700  000200                      MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>            ; 5107
 5378  006467' 431010  005340
 5379  006470' 511051  102004                      MWORD   <JMAP,J=5110,SOA,A=4,OR,D=1,OENA,SELE,MGC=4>  ; 5110
 5380  006471' 431040  005040
 5381  006472' 777777  777777                      -1
```

```

                                   ;#*********************************************************************
                                   ;* Test 24 - Fmtr Up Shift 8 (from PLIN) Test - SELRHTNIB=1
                                   ;
                                   ; Description:  Verify that the formatter can shift various  data
                                   ;               patterns up 4 bits at a time while inserting bits
                                   ;               from the PLIN into the formatter. The data loaded
                                   ;               into the PLIN initially is 360.
                                   ;
                                   ; Procedure:    KL> Port Clear
                                   ;               KL> Set up data transfer for data patterns
                                   ;
                                   ;               UC> Read a data pattern over the CBUS
                                   ;               UC> Load PLIN with 360
                                   ;
                                   ;               UC> Shift up 4 bits
                                   ;               UC> Read formatter contents and write into EBUF
                                   ;               KL> Read EBUF and verify the data
                                   ;               KL> Start up the ucode to shift again
                                   ;
                                   ;               Repeat the above shifting then verifying EBUF
                                   ;                 contents 10 times, until the data becomes
                                   ;                 all zero.
                                   ;
                                   ;               Do for data patterns:   0's
                                   ;                                       1's
                                   ;                                       525252,,525252
                                   ;                                       000000,,007777
                                   ;                                       000000,,007070
                                   ;                                       000000,,000707
                                   ;
                                   ; Failure:      ---
                                   ;#*********************************************************************

                                   ; Test data

006473' 254 00 0 00 006504'        TSTC24: JRST    TG24                    ; go start test
006474' 100604  000024                     CBUS!NDMP!NDCB!ZCBUS!24         ; test mask
006475' 006663' 013377'                    T24M,,[ASCIZ ^Fmtr Up Shift 4 (from PLIN) Test - SELRHTNIB=1^]
006476' 013263' 012716'                    [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
006477' 000000  006754'                    TSTC25                          ; failure test table
006500' 000000  007062'                    TSTC26                          ;   ...
006501' 000000  010224'                    TSTC31
006502' 000000  010542'                    TSTC32
006503' 777777  777777                     -1

                                   ; Start test

006504' 201 00 0 00 000000'        TG24:   MOVEI   Z9                      ; get address of module start
006505' 260 17 0 00 006226*                GO      TRACE                   ; handle trace output
006506' 201 01 0 00 006663'                MOVEI   1,T24M                  ; set up microcode address
006507' 260 17 0 00 006230*                GO      TLOAD                   ; load/verify it
006510' 263 17 0 00 000000                 RTN                             ; failed - exit test
```

```

                                          ; Initialization

006511' 400 15 0 00 000000        TL24:   SETZ    ERFLG,                  ; clear error flag
006512' 260 17 0 00 006233*               GO      IPACLR                  ; clear port
006513' 402 00 0 00 006234*               SETZM   TSTSUB                  ; initialize subtest number
                                          MOVSI   [000000,,000000         ; get address of buffer
                                                   777777,,777777         ;   contents
                                                   525252,,525252
                                                   000000,,007777
                                                   000000,,007070
006514' 205 00 0 00 013330'                        000000,,000707]
006515' 541 00 0 00 006251*               HRRI    BUFF                    ; build the BLT pointer
006516' 251 00 0 00 000000#               BLT     BUFF+5                  ; insert data into buffer

                                          ; First ensure the data transfer is reset

006517' 201 01 0 00 005000                MOVEI   1,5000                  ; get start address
006520' 242 01 0 00 000001                LSH     1,1                     ; position correctly
006521' 260 17 0 00 006351*               GO      LDRAR                   ; load the register
006522' 201 01 0 00 000010                MOVEI   1,MPRUN                 ; get MPRUN bit
006523' 260 17 0 00 006355*               GO      LDCSR                   ; start up the port
006524' 400 01 0 00 000000                SETZ    1,                      ; clear data
006525' 260 17 0 00 006523*               GO      LDCSR                   ; stop the port

                                          ; Now generate CCW list

006526' 200 01 0 00 013210'               MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
006527' 260 17 0 00 006250*               GO      CHINIT                  ; initialize software
006530' 551 01 0 00 006515*               HRRZI   1,BUFF                  ; buffer address
006531' 201 02 0 00 000006                MOVEI   2,6                     ; word count
006532' 400 03 0 00 000000                SETZ    3,                      ; clear zero fill specifier
006533' 260 17 0 00 006254*               GO      GENCCW                  ; generate a CCW list
006534' 201 06 0 00 006557'               MOVEI   6,TS24                  ; get test execute table address

                                          ; Start up the data transfer

006535' 201 01 0 00 001000                MOVEI   1,1000                  ; get start address
006536' 242 01 0 00 000001                LSH     1,1                     ; position correctly
006537' 260 17 0 00 006521*               GO      LDRAR                   ; load the register
006540' 201 01 0 00 000010                MOVEI   1,MPRUN                 ; get MPRUN bit
006541' 260 17 0 00 006525*               GO      LDCSR                   ; start up the port
006542' 201 00 0 00 000001                MOVEI   1                       ; set up delay of 1 ms
006543' 260 17 0 00 006264*               GO      ODELAY                  ; wait a bit
006544' 400 01 0 00 000000                SETZ    1,                      ; clear data
006545' 260 17 0 00 006541*               GO      LDCSR                   ; stop the port
```

```

                                  ; Loop on test execute table entries

006546' 260 17 0 00 006267*      TA24:   GO      TEXEC                   ; execute table entry
006547' 254 00 0 00 006556'              JRST    TX24                    ; end of table
006550' 254 00 0 00 006546'              JRST    TA24                    ; keep looping after call
006551' 474 15 0 00 000000               SETO    ERFLG,                  ; error - EBUF data incorrect

                                  ; Handle error printouts and scope looping

006552' 027 00 0 00 006650'              SCOPER  MA24                    ; print error message
006553' 254 00 0 00 006511'              JRST    TL24                    ; loop on error
006554' 254 00 0 00 006556'              JRST    TX24                    ; altmode exit
006555' 322 15 0 00 006546'              JUMPE   ERFLG,TA24              ; do next sstep table entry

                                  ; End of test

006556' 263 17 0 00 000000       TX24:   RTN                             ; return

                                  ; Test Execute Table, as:  (CMD,parameters)

006557' 140000  006607'          TS24:   TTABLE  (TCALL,TS24IN)          ; initialize stuff
006560' 200000  006612'          TS24L:  TTABLE  (TCALLC,TS24NE)         ; set up next data pattern

                                  ; Shifted 8 bits

006561' 140000  006640'          TS24A:  TTABLE  (TCALL,TS24NN)          ; increment shift count
006562' 400000  000100                   TTABLE  (TSSTAR,100)            ; start up port
006563' 240001  006643'                  TTABLE  (TCHECK,1,TS24RS)       ; check if completed
006564' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

                                  ; Shifted 16 bits

006565' 140000  006640'                  TTABLE  (TCALL,TS24NN)          ; increment shift count
006566' 400000  000100                   TTABLE  (TSSTAR,100)            ; start up port
006567' 240001  006644'                  TTABLE  (TCHECK,1,TS24RS+1)     ; check if completed
006570' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

                                  ; Shifted 24 bits

006571' 140000  006640'                  TTABLE  (TCALL,TS24NN)          ; increment shift count
006572' 400000  000100                   TTABLE  (TSSTAR,100)            ; start up port
006573' 240001  006645'                  TTABLE  (TCHECK,1,TS24RS+2)     ; check if completed
006574' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

                                  ; Shifted 32 bits

006575' 140000  006640'                  TTABLE  (TCALL,TS24NN)          ; increment shift count
006576' 400000  000100                   TTABLE  (TSSTAR,100)            ; start up port
006577' 240001  006646'                  TTABLE  (TCHECK,1,TS24RS+3)     ; check if completed
006600' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

```

```
                                        ; Shifted 40 bits

006601' 140000  006640'                         TTABLE  (TCALL,TS24NN)          ; increment shift count
006602' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
006603' 240001  006647'                         TTABLE  (TCHECK,1,TS24RS+4)     ; check if completed
006604' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet
006605' 300000  006560'                         TTABLE  (TJRST,TS24L)           ; loop till done
006606' 000000  000000                          TTABLE  (TLAST)

                                        ; Initialize stuff

006607' 201 00 0 00 000000#             TS24IN: MOVEI   BUFF-1                  ; initialize buffer pointer
006610' 202 00 0 00 014026'                     MOVEM   T24BUF#                 ;   ...
006611' 263 17 0 00 000000                      RTN                             ; return

                                        ; Set up next data pattern

006612' 350 01 0 00 014026'             TS24NE: AOS     1,T24BUF                ; point to next data pattern
006613' 307 01 0 00 000000#                     CAIG    1,BUFF+5                ; done yet?
006614' 350 00 0 17 000000                      AOS     (P)                     ; no - set up proper return
006615' 200 00 0 01 000000                      MOVE    (1)                     ; get data pattern
006616' 201 02 0 00 006643'                     MOVEI   2,TS24RS                ; get initial address of EBUF data
006617' 400 01 0 00 000000              TS24NO: SETZ    1,                      ; first shift doesn't include PLIN data
006620' 302 02 0 00 006643'                     CAIE    2,TS24RS                ;   so check if first shift and if not
006621' 505 01 0 00 741777                      HRLI    1,741777                ;   get initial PLIN data (left 8 bits)
006622' 246 00 0 00 000010                      LSHC    8                       ; shift 8 bits
006623' 202 00 0 02 000000                      MOVEM   (2)                     ; save it
006624' 350 00 0 00 000002                      AOS     2                       ; point to next address
006625' 307 02 0 00 006647'                     CAIG    2,TS24RS+4              ; done yet?
006626' 254 00 0 00 006617'                     JRST    TS24NO                  ; no - loop till done
006627' 402 00 0 00 014051'                     SETZM   TS24NM#                 ; initialize number of shifts

                                        ; Read the next word

006630' 201 01 0 00 000200                      MOVEI   1,200                   ; get start address
006631' 242 01 0 00 000001                      LSH     1,1                     ; position correctly
006632' 260 17 0 00 006537*                     GO      LDRAR                   ; load the register
006633' 201 01 0 00 000010                      MOVEI   1,MPRUN                 ; get MPRUN bit
006634' 260 17 0 00 006545*                     GO      LDCSR                   ; start up the port
006635' 400 01 0 00 000000                      SETZ    1,                      ; clear data
006636' 260 17 0 00 006634*                     GO      LDCSR                   ; stop the port
006637' 263 17 0 00 000000                      RTN                             ; return

                                        ; Increment shift count

006640' 201 00 0 00 000010              TS24NN: MOVEI   8                       ; get a 8
006641' 272 00 0 00 014051'                     ADDM    TS24NM                  ; add it
006642' 263 17 0 00 000000                      RTN                             ; return

006643'                                 TS24RS: BLOCK   5                       ; correct EBUF data
```

```

                                         ; Error messages

      006650' 240000  006652'            MA24:   CALL!TXNOT!MA24PN               ; print data pattern ...
      006651' 000000000000#                      LAST!CALL!TXALL!TTPNT           ; print test data

      006652' 037 00 0 00 013336'        MA24PN: TMSGC   <Data >                 ; get data and
      006653' 200 01 0 00 014026'                MOVE    1,T24BUF                ;   print it
      006654' 200 00 0 01 000000                 MOVE    (1)
      006655' 037 13 0 00 000000                 PNTHW
      006656' 037 00 0 00 013372'                TMSG    < shifted up from PLIN >
      006657' 200 00 0 00 014051'                MOVE    TS24NM                  ; get shift count
      006660' 037 15 0 00 000000                 PNTDEC
      006661' 037 00 0 00 013343'                TMSG    <. bits, result in EBUF>
      006662' 263 17 0 00 000000                 RTN

                                         ; Microcode - Shift 8 bits

      006663' 010001  010000             T24M:   MWORD   <ADDR=100,JMAP,J=101,D=1>                        ; 100
      006664' 001000  000040
      006665' 010100  000222                     MWORD   <CONT,D=1,SELF,MGC=222>                          ; 101
      006666' 001000  003340
      006667' 010200  000200                     MWORD   <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>               ; 102
      006670' 732002  002340
      006671' 010351  000000                     MWORD   <JMAP,J=5100,D=1>                                ; 103
      006672' 001000  000040

                                         ;       Routine to read a word from CBUS into the formatter.

      006673' 020002  010000                     MWORD   <ADDR=200,JMAP,J=201>                            ; 200
      006674' 001000  000040
      006675' 020102  030000                     MWORD   <CJP,J=203,D=1,CENA,CCCA>                        ; 201
      006676' 001400  000060
      006677' 020202  010000                     MWORD   <JMAP,J=201,D=1>                                 ; 202
      006700' 001000  000040
      006701' 020300  000004                     MWORD   <CONT,D=1,SELC,MGC=4>                            ; 203
      006702' 001000  004340
      006703' 020400  000200                     MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>               ; 204
      006704' 732001  402340
      006705' 020500  000000                     MWORD   <JZ>                                             ; 205
      006706' 000000  000000

                                         ;       Initialization

      006707' 100010  010020                     MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>              ; 1000
      006710' 000000  002040
      006711' 100100  000200                     MWORD   <CONT,D=1,SELC,MGC=200>                          ; 1001
      006712' 001000  004340
      006713' 100200  310360                     MWORD   <LDCT,J=31,SDO,OR,D=2,SKCN,MGC=360>              ; 1002
      006714' 732000  240300
      006715' 100310  030000                     MWORD   <RPCT,J=1003,SOA,OR,D=5>                         ; 1003
      006716' 435000  000220
      006717' 100400  002040                     MWORD   <CONT,SOA,OR,D=1,OENA,SELM,MGC=40>               ; 1004
      006720' 431000  002340
      006721' 100500  000006                     MWORD   <JZ,SDO,D=1,SELP,MGC=6>                          ; 1005
```

```
006722' 701000  001000
006723' 000000  000000                  MWORD   <ADDR=0,JZ>                                     ; 0
006724' 000000  000000

                                ;       Initialization step

006725' 500050  010100                  MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>            ; 5000
006726' 000000  004040
006727' 500100  000020                  MWORD   <JZ,SELM,MGC=20>                                ; 5001
006730' 000000  002000

                                ; Subroutine - Passed - Write 1 to bit 12 of CSR register

006731' 510000  000000                  MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>                ; 5100
006732' 742001  000340
006733' 510100  000000                  MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                     ; 5101
006734' 302001  000740
006735' 510200  260000                  MWORD   <LDCT,J=26>                                     ; 5102
006736' 000000  000300
006737' 510351  030000                  MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>              ; 5103
006740' 102021  000220
006741' 510400  002004                  MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>           ; 5104
006742' 431020  005340
006743' 510551  070040                  MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>          ; 5105
006744' 001400  015060
006745' 510651  050000                  MWORD   <JMAP,J=5105,D=1>                               ; 5106
006746' 001000  000040
006747' 510700  000200                  MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>              ; 5107
006750' 431010  005340
006751' 511051  102004                  MWORD   <JMAP,J=5110,SOA,A=4,OR,D=1,OENA,SELE,MGC=4>    ; 5110
006752' 431040  005040
006753' 777777  777777                  -1
```

```

                                        ;#*********************************************************************
                                        ;* Test 25 - Fmtr Basic Down Shifting Test
                                        ;
                                        ; Description:  Verify that the  Mover/Formatter  is  capable  of
                                        ;               being shifted down. The contents of the formatter
                                        ;               is initially zero, and the result of any shifting
                                        ;               should be zero.
                                        ;
                                        ; Procedure:    KL> Port Clear
                                        ;               UC> Shift formatter 4 bits down
                                        ;               UC> Read formatter data
                                        ;               UC> If zero, continue.
                                        ;                   If non-zero, halt with a CRAM PE and leave
                                        ;                     the non-zero data read in the EBUF.
                                        ;               KL> Let it run at full speed for 1/2 second then
                                        ;                   halt and verify that no CRAM PE occurred and
                                        ;                   that the microcode did not stop at the error
                                        ;                   location.
                                        ;
                                        ; Failure:      ---
                                        ;#*********************************************************************

                                        ; Test data

 006754' 254 00 0 00 006764'            TSTC25: JRST    TG25                    ; go start test
 006755' 100604  000025                         CBUS!NDMP!NDCB!ZCBUS!25         ; test mask
 006756' 007047' 013435'                        T25M,,[ASCIZ ^Fmtr Basic Down Shifting Test^]
 006757' 013263' 012716'                        [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
 006760' 000000  007062'                        TSTC26                          ; failure test table
 006761' 000000  007372'                        TSTC27                          ;   ...
 006762' 000000  007710'                        TSTC30
 006763' 777777  777777                         -1

                                        ; Start test

 006764' 201 00 0 00 000000'            TG25:   MOVEI   Z9                      ; get address of module start
 006765' 260 17 0 00 006505*                    GO      TRACE                   ; handle trace output
 006766' 201 01 0 00 007047'                    MOVEI   1,T25M                  ; set up microcode address
 006767' 260 17 0 00 006507*                    GO      TLOAD                   ; load/verify it
 006770' 263 17 0 00 000000                     RTN                             ; failed - exit test
```

```

                                 ; 1st segment of test (Segment A) - Start up port

006771' 400 15 0 00 000000       TA25:    SETZ    ERFLG,                 ; clear error flag
006772' 201 00 0 00 000001                MOVEI   1                      ; initialize subtest number
006773' 202 00 0 00 006513*               MOVEM   TSTSUB                 ;  to 1
006774' 260 17 0 00 006512*               GO      IPACLR                 ; issue a port clear
006775' 402 00 0 00 003123*               SETZM   SNEXT                  ; set up start location
006776' 201 00 0 00 000010                MOVEI   10                     ; get a 10
006777' 202 00 0 00 003125*               MOVEM   SDATA                  ; set up start CSR data
007000' 260 17 0 00 003126*               GO      IPASRT                 ; start up the port
007001' 474 15 0 00 000000                SETO    ERFLG,                 ; failed - error reading/writing CSR
007002' 474 15 0 00 000000                SETO    ERFLG,                 ; failed - port already running
007003' 474 15 0 00 000000                SETO    ERFLG,                 ; failed - error bits set in CSR
007004' 027 00 0 00 007027'               SCOPER  MA25                   ; print error message
007005' 254 00 0 00 006771'               JRST    TA25                   ; loop on error
007006' 254 00 0 00 007026'               JRST    TX25                   ; altmode exit
007007' 326 15 0 00 007026'               JUMPN   ERFLG,TX25             ; error already - exit test

                                 ; 2nd segment of test (Segment B) - Check if CRAM PE sets after 1/2 second

007010' 350 00 0 00 006773*               AOS     TSTSUB                 ; increment subtest number
007011' 201 00 0 00 000764                MOVEI   ^D500                  ; wait 1/2 second
007012' 260 17 0 00 006543*               GO      ODELAY                 ; do the wait
007013' 260 17 0 00 003157*               GO      RDCSR                  ; read CSR
007014' 474 15 0 00 000000                SETO    ERFLG,                 ; failed - couldn't read CSR
007015' 603 01 0 00 004000                TLNE    1,(CRAMPE)             ; got a CRAM PE?
007016' 474 15 0 00 000000                SETO    ERFLG,                 ; yes - set error flag
007017' 201 01 0 00 204000                MOVEI   1,EBUSPE!TSTEBF        ; get bits to clear error condition
007020' 260 17 0 00 006636*               GO      LDCSR                  ;   and set up to read the EBUF
007021' 260 17 0 00 003147*               GO      RDEBUF                 ; read EBUF data
007022' 202 01 0 00 014013'               MOVEM   1,SAVDAT#              ; save data
007023' 027 00 0 00 007041'               SCOPER  MB25                   ; print error message
007024' 254 00 0 00 006771'               JRST    TA25                   ; loop on error
007025' 254 00 0 00 007026'               JRST    TX25                   ; altmode exit

007026' 263 17 0 00 000000       TX25:    RTN                            ; exit test

                                 ; Error messages

007027' 140000  013266'          MA25:    MSG!TXNOT![ASCIZ /Couldn't successfully start the port/]
007030' 270000  007031'                   LAST!CALL!TXALL!MA25P1

007031' 260 17 0 00 007013*      MA25P1:  GO      RDCSR                  ; get CSR contents
007032' 255 00 0 00 000000                JFCL                           ; error
007033' 202 01 0 00 014010'               MOVEM   1,SAVCSR#              ; save it
007034' 200 01 0 00 014010'               MOVE    1,SAVCSR               ; get CSR data
007035' 201 02 0 00 000023                MOVEI   2,^D19                 ; character count so far
007036' 201 03 0 00 000023                MOVEI   3,^D19                 ; continuation line column number
007037' 260 17 0 00 003165*               GO      CSRPNT                 ; go print in English
007040' 263 17 0 00 000000                RTN                            ; return
```

```

007041' 140000  013443'         MB25:   MSG!TXNOT![ASCIZ /Down shifting of zero data resulted in non-zero data/]
007042' 270000  007043'                 LAST!CALL!TXALL!MA25PN

007043' 037 00 0 00 013311'     MA25PN: TMSGC   <EBUF Data (Formatter contents):  >
007044' 200 00 0 00 014013'             MOVE    SAVDAT
007045' 037 13 0 00 000000              PNTHW
007046' 263 17 0 00 000000              RTN

                                ; Microcode:

007047' 000000  010104          T25M:   MWORD   <ADDR=0,JMAP,J=1,SOA,D=1,SELF,MGC=104>  ; 0
007050' 401000  003040
007051' 000100  000200                  MWORD   <CONT,SDO,OR,D=2,SELM,MGC=200>          ; 1
007052' 732000  002340
007053' 000200  000000                  MWORD   <CJP,J=0,SOA,OR,D=1,CENA,CCFZ>          ; 2
007054' 431400  020060
007055' 000300  002004                  MWORD   <CONT,SOA,OR,D=1,OENA,SELE,MGC=4>       ; 3
007056' 431000  005340
007057' 000400  040000                  MWORD   <JMAP,J=4,BAD>                          ; 4
007060' 000000  000041
007061' 777777  777777                  -1
```

```

                                        ;#**************************************************************************
                                        ;* Test 26 - Fmtr Down Shift 4 Test
                                        ;
                                        ; Description:  Verify that the formatter can shift various  data
                                        ;               patterns down 4 bits at a time.
                                        ;
                                        ; Procedure:    KL> Port Clear
                                        ;               KL> Set up data transfer for data patterns
                                        ;
                                        ;               UC> Read a data pattern over the CBUS
                                        ;               UC> Shift down 4 bits
                                        ;               UC> Read formatter contents and write into
                                        ;                   EBUF
                                        ;               KL> Read EBUF and verify the data
                                        ;               KL> Start up the ucode to shift again
                                        ;
                                        ;               Repeat the above shifting then verifying EBUF
                                        ;                 contents 10 times, until the data becomes
                                        ;                 all zero.
                                        ;
                                        ;               Do for data patterns:   0's
                                        ;                                       1's
                                        ;                                       525252,,525252
                                        ;                                       000000,,007777
                                        ;                                       000000,,007070
                                        ;                                       000000,,000707
                                        ;
                                        ; Failure:      ---
                                        ;#**************************************************************************

                                        ; Test data

007062' 254 00 0 00 007073'             TSTC26: JRST    TG26                    ; go start test
007063' 100604  000026                          CBUS!NDMP!NDCB!ZCBUS!26         ; test mask
007064' 007305' 013456'                         T26M,,[ASCIZ ^Fmtr Down Shift 4 Test^]
007065' 013263' 012716'                         [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
007066' 000000  007372'                         TSTC27                          ; failure test table
007067' 000000  007710'                         TSTC30                          ;   ...
007070' 000000  010224'                         TSTC31
007071' 000000  010542'                         TSTC32
007072' 777777  777777                          -1

                                        ; Start test

007073' 201 00 0 00 000000'             TG26:   MOVEI   Z9                      ; get address of module start
007074' 260 17 0 00 006765*                     GO      TRACE                   ; handle trace output
007075' 201 01 0 00 007305'                     MOVEI   1,T26M                  ; set up microcode address
007076' 260 17 0 00 006767*                     GO      TLOAD                   ; load/verify it
007077' 263 17 0 00 000000                      RTN                             ; failed - exit test
```

```

                                       ; Initialization

007100' 400 15 0 00 000000             TL26:    SETZ    ERFLG,                   ; clear error flag
007101' 260 17 0 00 006774*                     GO      IPACLR                   ; clear port
007102' 402 00 0 00 007010*                     SETZM   TSTSUB                   ; initialize subtest number
                                                MOVSI   [000000,,000000          ; get address of buffer
                                                         777777,,777777          ;   contents
                                                         525252,,525252
                                                         000000,,007777
                                                         000000,,007070
007103' 205 00 0 00 013330'                              000000,,000707]
007104' 541 00 0 00 006530*                     HRRI    BUFF                     ; build the BLT pointer
007105' 251 00 0 00 000000#                     BLT     BUFF+5                   ; insert data into buffer

                                       ; First ensure the data transfer is reset

007106' 201 01 0 00 005000                      MOVEI   1,5000                   ; get start address
007107' 242 01 0 00 000001                      LSH     1,1                      ; position correctly
007110' 260 17 0 00 006632*                     GO      LDRAR                    ; load the register
007111' 201 01 0 00 000010                      MOVEI   1,MPRUN                  ; get 'MPRUN' bit
007112' 260 17 0 00 007020*                     GO      LDCSR                    ; start up the port
007113' 400 01 0 00 000000                      SETZ    1,                       ; stop the port
007114' 260 17 0 00 007112*                     GO      LDCSR

                                       ; Now generate CCW list

007115' 200 01 0 00 013210'                     MOVE    1,[CBUFSZ,,CBUF]         ; get pointer to control buffer
007116' 260 17 0 00 006527*                     GO      CHINIT                   ; initialize software
007117' 551 01 0 00 007104*                     HRRZI   1,BUFF                   ; buffer address
007120' 201 02 0 00 000006                      MOVEI   2,6                      ; word count
007121' 400 03 0 00 000000                      SETZ    3,                       ; clear zero fill specifier
007122' 260 17 0 00 006533*                     GO      GENCCW                   ; generate a CCW list
007123' 201 06 0 00 007146'                     MOVEI   6,TS26                   ; get test execute table address

                                       ; Start up the data transfer

007124' 201 01 0 00 001000                      MOVEI   1,1000                   ; get start address
007125' 242 01 0 00 000001                      LSH     1,1                      ; position correctly
007126' 260 17 0 00 007110*                     GO      LDRAR                    ; load the register
007127' 201 01 0 00 000010                      MOVEI   1,MPRUN                  ; get MPRUN bit
007130' 260 17 0 00 007114*                     GO      LDCSR                    ; start up the port
007131' 201 00 0 00 000001                      MOVEI   1                        ; set up delay of 1 ms
007132' 260 17 0 00 007012*                     GO      ODELAY                   ; wait a bit
007133' 400 01 0 00 000000                      SETZ    1,                       ; clear data
007134' 260 17 0 00 007130*                     GO      LDCSR                    ; stop the port
```

```

                                        ; Loop on test execute table entries

007135' 260 17 0 00 006546*             TA26:   GO      TEXEC                   ; execute table entry
007136' 254 00 0 00 007145'                     JRST    TX26                    ; end of table
007137' 254 00 0 00 007135'                     JRST    TA26                    ; keep looping after call
007140' 474 15 0 00 000000                      SETO    ERFLG,                  ; error - EBUF data incorrect

                                        ; Handle error printouts and scope looping

007141' 027 00 0 00 007272'                     SCOPER  MA26                    ; print error message
007142' 254 00 0 00 007100'                     JRST    TL26                    ; loop on error
007143' 254 00 0 00 007145'                     JRST    TX26                    ; altmode exit
007144' 322 15 0 00 007135'                     JUMPE   ERFLG,TA26              ; do next sstep table entry

                                        ; End of test

007145' 263 17 0 00 000000              TX26:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

007146' 140000  007222'                 TS26:   TTABLE  (TCALL,TS26IN)          ; initialize stuff
007147' 200000  007225'                 TS26L:  TTABLE  (TCALLC,TS26NE)         ; set up next data pattern

                                        ; Shifted 4 bits

007150' 140000  007255'                 TS26A:  TTABLE  (TCALL,TS26NN)          ; increment shift count
007151' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
007152' 240001  007260'                         TTABLE  (TCHECK,1,TS26RS)       ; check if completed
007153' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 8 bits

007154' 140000  007255'                         TTABLE  (TCALL,TS26NN)          ; increment shift count
007155' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
007156' 240001  007261'                         TTABLE  (TCHECK,1,TS26RS+1)     ; check if completed
007157' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 12 bits

007160' 140000  007255'                         TTABLE  (TCALL,TS26NN)          ; increment shift count
007161' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
007162' 240001  007262'                         TTABLE  (TCHECK,1,TS26RS+2)     ; check if completed
007163' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 16 bits

007164' 140000  007255'                         TTABLE  (TCALL,TS26NN)          ; increment shift count
007165' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
007166' 240001  007263'                         TTABLE  (TCHECK,1,TS26RS+3)     ; check if completed
007167' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

```

```

                                ; Shifted 20 bits

007170' 140000  007255'                 TTABLE  (TCALL,TS26NN)          ; increment shift count
007171' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
007172' 240001  007264'                 TTABLE  (TCHECK,1,TS26RS+4)     ; check if completed
007173' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                ; Shifted 24 bits

007174' 140000  007255'                 TTABLE  (TCALL,TS26NN)          ; increment shift count
007175' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
007176' 240001  007265'                 TTABLE  (TCHECK,1,TS26RS+5)     ; check if completed
007177' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                ; Shifted 28 bits

007200' 140000  007255'                 TTABLE  (TCALL,TS26NN)          ; increment shift count
007201' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
007202' 240001  007266'                 TTABLE  (TCHECK,1,TS26RS+6)     ; check if completed
007203' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                ; Shifted 32 bits

007204' 140000  007255'                 TTABLE  (TCALL,TS26NN)          ; increment shift count
007205' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
007206' 240001  007267'                 TTABLE  (TCHECK,1,TS26RS+7)     ; check if completed
007207' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                ; Shifted 36 bits

007210' 140000  007255'                 TTABLE  (TCALL,TS26NN)          ; increment shift count
007211' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
007212' 240001  007270'                 TTABLE  (TCHECK,1,TS26RS+10)    ; check if completed
007213' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                ; Shifted 40 bits

007214' 140000  007255'                 TTABLE  (TCALL,TS26NN)          ; increment shift count
007215' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
007216' 240001  007271'                 TTABLE  (TCHECK,1,TS26RS+11)    ; check if completed
007217' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet
007220' 300000  007147'                 TTABLE  (TJRST,TS26L)           ; loop till done
007221' 000000  000000                  TTABLE  (TLAST)

                                ; Initialize stuff

007222' 201 00 0 00 000000#     TS26IN: MOVEI   BUFF-1                  ; initialize buffer pointer
007223' 202 00 0 00 014027'             MOVEM   T26BUF#                 ;  ...
007224' 263 17 0 00 000000              RTN                             ; return
```

```

                                        ; Set up next data pattern

007225' 350 01 0 00 014027'     TS26NE: AOS     1,T26BUF                ; point to next data pattern
007226' 307 01 0 00 000000#             CAIG    1,BUFF+5                ; done yet?
007227' 350 00 0 17 000000              AOS     (P)                     ; no - set up proper return
007230' 200 00 0 01 000000              MOVE    (1)                     ; get data pattern
007231' 400 01 0 00 000000              SETZ    1,                      ; clear word to shift into
007232' 201 02 0 00 007260'             MOVEI   2,TS26RS                ; get initial address of EBUF data
007233' 246 00 0 00 777774      TS26NO: LSHC    -4                      ; shift 4 bits into AC1
007234' 135 03 0 00 013463'             LDB     3,[POINT 4,1,15]        ; get 4 bits
007235' 137 03 0 00 013464'             DPB     3,[POINT 4,0,3]         ; save them
007236' 202 00 0 02 000000              MOVEM   (2)                     ; save it
007237' 350 00 0 00 000002              AOS     2                       ; point to next address
007240' 307 02 0 00 007271'             CAIG    2,TS26RS+11             ; done yet?
007241' 254 00 0 00 007233'             JRST    TS26NO                  ; no - loop till done
007242' 402 00 0 00 014052'             SETZM   TS26NM#                 ; initialize number of shifts

                                ; Read the next word

007243' 400 01 0 00 000000              SETZ    1,                      ; clear data
007244' 260 17 0 00 007134*             GO      LDCSR                   ; stop the port
007245' 201 01 0 00 000200              MOVEI   1,200                   ; get start address
007246' 242 01 0 00 000001              LSH     1,1                     ; position correctly
007247' 260 17 0 00 007126*             GO      LDRAR                   ; load the register
007250' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
007251' 260 17 0 00 007244*             GO      LDCSR                   ; start up the port
007252' 400 01 0 00 000000              SETZ    1,                      ; clear data
007253' 260 17 0 00 007251*             GO      LDCSR                   ; stop the port
007254' 263 17 0 00 000000              RTN                             ; return

                                ; Increment shift count

007255' 201 00 0 00 000004      TS26NN: MOVEI   4                       ; get a 4
007256' 272 00 0 00 014052'             ADDM    TS26NM                  ; add it
007257' 263 17 0 00 000000              RTN                             ; return

007260'                         TS26RS: BLOCK   12                      ; correct EBUF data

                                ; Error messages

007272' 240000  007274'         MA26:   CALL!TXNOT!MA26PN               ; print data pattern ...
007273' 000000000000#                   LAST!CALL!TXALL!TTPNT           ; print test data

007274' 037 00 0 00 013336'     MA26PN: TMSGC   <Data >                 ; get data and
007275' 200 01 0 00 014027'             MOVE    1,T26BUF                ;   print it
007276' 200 00 0 01 000000              MOVE    (1)
007277' 037 13 0 00 000000              PNTHW
007300' 037 00 0 00 013465'             TMSG    < shifted down >
007301' 200 00 0 00 014052'             MOVE    TS26NM                  ; get shift count
007302' 037 15 0 00 000000              PNTDEC
007303' 037 00 0 00 013343'             TMSG    <. bits, result in EBUF>
007304' 263 17 0 00 000000              RTN
```

```

                                ; Microcode - Shift 4 bits

007305' 010001  010000          T26M:   MWORD   <ADDR=100,JMAP,J=101,D=1>                       ; 100
007306' 001000  000040
007307' 010100  000104                  MWORD   <CONT,D=1,SELF,MGC=104>                         ; 101
007310' 001000  003340
007311' 010200  000200                  MWORD   <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>              ; 102
007312' 732002  002340
007313' 010351  000000                  MWORD   <JMAP,J=5100,D=1>                               ; 103
007314' 001000  000040

                                ;       Routine to read a word from CBUS into the formatter.

007315' 020002  010000                  MWORD   <ADDR=200,JMAP,J=201>                           ; 200
007316' 000000  000040
007317' 020102  030000                  MWORD   <CJP,J=203,D=1,CENA,CCCA>                       ; 201
007320' 001400  000060
007321' 020202  010000                  MWORD   <JMAP,J=201,D=1>                                ; 202
007322' 001000  000040
007323' 020300  000200                  MWORD   <CONT,D=1,SELF,MGC=200>                         ; 203
007324' 001000  003340
007325' 020400  000204                  MWORD   <CONT,D=1,SELF,MGC=204>                         ; 204
007326' 001000  003340
007327' 020500  000004                  MWORD   <CONT,D=1,SELC,MGC=4>                           ; 205
007330' 001000  004340
007331' 020600  000200                  MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>              ; 206
007332' 732001  402340
007333' 020700  000000                  MWORD   <JZ>                                            ; 207
007334' 000000  000000

                                ;       Initialization

007335' 100010  010020                  MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>             ; 1000
007336' 000000  002040
007337' 100100  000200                  MWORD   <JZ,D=1,SELC,MGC=200>                           ; 1001
007340' 001000  004000
007341' 000000  000000                  MWORD   <ADDR=0,JZ>                                     ; 0
007342' 000000  000000

                                ;       Initialization step

007343' 500050  010100                  MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>            ; 5000
007344' 000000  004040
007345' 500100  000020                  MWORD   <JZ,SELM,MGC=20>                                ; 5001
007346' 000000  002000

                                ; Subroutine - Passed - Write 1 to bit 12 of CSR register

007347' 510000  000000                  MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>                ; 5100
007350' 742001  000340
007351' 510100  000000                  MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                     ; 5101
007352' 302001  000740
007353' 510200  260000                  MWORD   <LDCT,J=26>                                     ; 5102
007354' 000000  000300
```

```
  6091  007355' 510351  030000                  MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>                  ; 5103
  6092  007356' 102021  000220
  6093  007357' 510400  002004                  MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>               ; 5104
  6094  007360' 431020  005340
  6095  007361' 510551  070040                  MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>              ; 5105
  6096  007362' 001400  015060
  6097  007363' 510651  050000                  MWORD   <JMAP,J=5105,D=1>                                   ; 5106
  6098  007364' 001000  000040
  6099  007365' 510700  000200                  MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>                  ; 5107
  6100  007366' 431010  005340
  6101  007367' 511051  102004                  MWORD   <JMAP,J=5110,SOA,OR,A=4,D=1,OENA,SELE,MGC=4>        ; 5110
  6102  007370' 431040  005040
  6103  007371' 777777  777777                  -1
```

```

                                ;#********************************************************************
                                ;* Test 27 - Fmtr Down Shift 4 (from PLIN) Test - SELRHTNIB=0
                                ;
                                ; Description:  Verify that the formatter can shift various  data
                                ;               patterns 4 bits at a time  while  inserting bytes
                                ;               from the PLIN into the formatter. The data loaded
                                ;               into the PLIN initially is 017.
                                ;
                                ; Procedure:    KL> Port Clear
                                ;               KL> Set up data transfer for data patterns
                                ;
                                ;               UC> Read a data pattern over the CBUS
                                ;               UC> Load PLIN with 017
                                ;
                                ;               UC> Shift down 4 bits
                                ;               UC> Read formatter contents and write into EBUF
                                ;               KL> Read EBUF and verify the data
                                ;               KL> Start up the ucode to shift again
                                ;
                                ;               Repeat the above shifting then verifying EBUF
                                ;                 contents 10 times, until the data becomes
                                ;                 all zero.
                                ;
                                ;               Do for data patterns:   0's
                                ;                                       1's
                                ;                                       525252,,525252
                                ;                                       000000,,007777
                                ;                                       000000,,007070
                                ;                                       000000,,000707
                                ;
                                ; Failure:      ---
                                ;#********************************************************************

                                ; Test data

007372' 254 00 0 00 007404'     TSTC27: JRST    TG27                    ; go start test
007373' 100604  000027                  CBUS!NDMP!NDCB!ZCBUS!27         ; test mask
007374' 007613' 013470'                 T27M,,[ASCIZ ^Fmtr Down Shift 4 (from PLIN) Test - SELRHTNIB=0^]
007375' 013263' 012716'                 [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
007376' 000000  007710'                 TSTC30                          ; failure test table
007377' 000000  010224'                 TSTC31                          ;   ...
007400' 000000  010542'                 TSTC32
007401' 000000  011060'                 TSTC33
007402' 000000  011344'                 TSTC34
007403' 777777  777777                  -1

                                ; Start test

007404' 201 00 0 00 000000'     TG27:   MOVEI   Z9                      ; get address of module start
007405' 260 17 0 00 007074*             GO      TRACE                   ; handle trace output
007406' 201 01 0 00 007613'             MOVEI   1,T27M                  ; set up microcode address
007407' 260 17 0 00 007076*             GO      TLOAD                   ; load/verify it
007410' 263 17 0 00 000000              RTN                             ; failed - exit test
```

```

                                        ; Initialization

007411' 400 15 0 00 000000      TL27:   SETZ    ERFLG,                  ; clear error flag
007412' 260 17 0 00 007101*             GO      IPACLR                  ; clear port
007413' 402 00 0 00 007102*             SETZM   TSTSUB                  ; initialize subtest number
                                        MOVSI   [000000,,000000         ; get address of buffer
                                                 777777,,777777         ;   contents
                                                 525252,,525252
                                                 000000,,007777
                                                 000000,,007070
007414' 205 00 0 00 013330'                      000000,,000707]
007415' 541 00 0 00 007117*             HRRI    BUFF                    ; build the BLT pointer
007416' 251 00 0 00 000000#             BLT     BUFF+5                  ; insert data into buffer

                                        ; First ensure the data transfer is reset

007417' 201 01 0 00 005000              MOVEI   1,5000                  ; get start address
007420' 242 01 0 00 000001              LSH     1,1                     ; position correctly
007421' 260 17 0 00 007247*             GO      LDRAR                   ; load the register
007422' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
007423' 260 17 0 00 007253*             GO      LDCSR                   ; start up the port
007424' 400 01 0 00 000000              SETZ    1,                      ; clear data
007425' 260 17 0 00 007423*             GO      LDCSR                   ; stop the port

                                        ; Now generate CCW list

007426' 200 01 0 00 013210'             MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
007427' 260 17 0 00 007116*             GO      CHINIT                  ; initialize software
007430' 551 01 0 00 007415*             HRRZI   1,BUFF                  ; buffer address
007431' 201 02 0 00 000006              MOVEI   2,6                     ; word count
007432' 400 03 0 00 000000              SETZ    3,                      ; clear zero fill specifier
007433' 260 17 0 00 007122*             GO      GENCCW                  ; generate a CCW list
007434' 201 06 0 00 007457'             MOVEI   6,TS27                  ; get test execute table address

                                        ; Start up the data transfer

007435' 201 01 0 00 001000              MOVEI   1,1000                  ; get start address
007436' 242 01 0 00 000001              LSH     1,1                     ; position correctly
007437' 260 17 0 00 007421*             GO      LDRAR                   ; load the register
007440' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
007441' 260 17 0 00 007425*             GO      LDCSR                   ; start up the port
007442' 201 00 0 00 000001              MOVEI   1                       ; set up delay of 1 ms
007443' 260 17 0 00 007132*             GO      ODELAY                  ; wait a bit
007444' 400 01 0 00 000000              SETZ    1,                      ; clear data
007445' 260 17 0 00 007441*             GO      LDCSR                   ; stop the port

                                        ; Loop on test execute table entries

007446' 260 17 0 00 007135*     TA27:   GO      TEXEC                   ; execute table entry
007447' 254 00 0 00 007456'             JRST    TX27                    ; end of table
007450' 254 00 0 00 007446'             JRST    TA27                    ; keep looping after call
007451' 474 15 0 00 000000              SETO    ERFLG,                  ; error - EBUF data incorrect
```

```

                                        ; Handle error printouts and scope looping

007452' 027 00 0 00 007600'                   SCOPER  MA27                    ; print error message
007453' 254 00 0 00 007411'                   JRST    TL27                    ; loop on error
007454' 254 00 0 00 007456'                   JRST    TX27                    ; altmode exit
007455' 322 15 0 00 007446'                   JUMPE   ERFLG,TA27              ; do next sstep table entry

                                        ; End of test

007456' 263 17 0 00 000000              TX27:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

007457' 140000  007533'                 TS27:   TTABLE  (TCALL,TS27IN)          ; initialize stuff
007460' 200000  007536'                 TS27L:  TTABLE  (TCALLC,TS27NE)         ; set up next data pattern

                                        ; Shifted 4 bits

007461' 140000  007563'                 TS27A:  TTABLE  (TCALL,TS27NN)          ; increment shift count
007462' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
007463' 240001  007566'                         TTABLE  (TCHECK,1,TS27RS)       ; check if completed
007464' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 8 bits

007465' 140000  007563'                         TTABLE  (TCALL,TS27NN)          ; increment shift count
007466' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
007467' 240001  007567'                         TTABLE  (TCHECK,1,TS27RS+1)     ; check if completed
007470' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 12 bits

007471' 140000  007563'                         TTABLE  (TCALL,TS27NN)          ; increment shift count
007472' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
007473' 240001  007570'                         TTABLE  (TCHECK,1,TS27RS+2)     ; check if completed
007474' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 16 bits

007475' 140000  007563'                         TTABLE  (TCALL,TS27NN)          ; increment shift count
007476' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
007477' 240001  007571'                         TTABLE  (TCHECK,1,TS27RS+3)     ; check if completed
007500' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 20 bits

007501' 140000  007563'                         TTABLE  (TCALL,TS27NN)          ; increment shift count
007502' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
007503' 240001  007572'                         TTABLE  (TCHECK,1,TS27RS+4)     ; check if completed
007504' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

```

```

                                    ; Shifted 24 bits

007505' 140000  007563'                     TTABLE  (TCALL,TS27NN)          ; increment shift count
007506' 400000  000100                      TTABLE  (TSSTAR,100)            ; start up port
007507' 240001  007573'                     TTABLE  (TCHECK,1,TS27RS+5)     ; check if completed
007510' 340000  000000                      TTABLE  (TEXIT)                 ; exit if error yet

                                    ; Shifted 28 bits

007511' 140000  007563'                     TTABLE  (TCALL,TS27NN)          ; increment shift count
007512' 400000  000100                      TTABLE  (TSSTAR,100)            ; start up port
007513' 240001  007574'                     TTABLE  (TCHECK,1,TS27RS+6)     ; check if completed
007514' 340000  000000                      TTABLE  (TEXIT)                 ; exit if error yet

                                    ; Shifted 32 bits

007515' 140000  007563'                     TTABLE  (TCALL,TS27NN)          ; increment shift count
007516' 400000  000100                      TTABLE  (TSSTAR,100)            ; start up port
007517' 240001  007575'                     TTABLE  (TCHECK,1,TS27RS+7)     ; check if completed
007520' 340000  000000                      TTABLE  (TEXIT)                 ; exit if error yet

                                    ; Shifted 36 bits

007521' 140000  007563'                     TTABLE  (TCALL,TS27NN)          ; increment shift count
007522' 400000  000100                      TTABLE  (TSSTAR,100)            ; start up port
007523' 240001  007576'                     TTABLE  (TCHECK,1,TS27RS+10)    ; check if completed
007524' 340000  000000                      TTABLE  (TEXIT)                 ; exit if error yet

                                    ; Shifted 40 bits

007525' 140000  007563'                     TTABLE  (TCALL,TS27NN)          ; increment shift count
007526' 400000  000100                      TTABLE  (TSSTAR,100)            ; start up port
007527' 240001  007577'                     TTABLE  (TCHECK,1,TS27RS+11)    ; check if completed
007530' 340000  000000                      TTABLE  (TEXIT)                 ; exit if error yet
007531' 300000  007460'                     TTABLE  (TJRST,TS27L)           ; loop till done
007532' 000000  000000                      TTABLE  (TLAST)

                                    ; Initialize stuff

007533' 201 00 0 00 000000#         TS27IN: MOVEI   BUFF-1                  ; initialize buffer pointer
007534' 202 00 0 00 014030'                 MOVEM   T27BUF#                 ; ...
007535' 263 17 0 00 000000                  RTN                             ; return
```

```

                                ; Set up next data pattern

007536' 350 01 0 00 014030'     TS27NE: AOS     1,T27BUF                ; point to next data pattern
007537' 307 01 0 00 000000#             CAIG    1,BUFF+5                ; done yet?
007540' 350 00 0 17 000000              AOS     (P)                     ; no - set up proper return
007541' 200 00 0 01 000000              MOVE    (1)                     ; get data pattern
007542' 201 02 0 00 007566'             MOVEI   2,TS27RS                ; get initial address of EBUF data
007543' 242 00 0 00 777774      TS27NO: LSH     -4                      ; shift 4 bits
007544' 303 02 0 00 007567'             CAILE   2,TS27RS+1              ; if past the first 2 shifts, insert
007545' 661 00 0 00 740000              TLO     740000                  ;   the PLIN data into left 4 bits
007546' 202 00 0 02 000000              MOVEM   (2)                     ; save it
007547' 350 00 0 00 000002              AOS     2                       ; point to next address
007550' 307 02 0 00 007577'             CAIG    2,TS27RS+11             ; done yet?
007551' 254 00 0 00 007543'             JRST    TS27NO                  ; no - loop till done
007552' 402 00 0 00 014053'             SETZM   TS27NM#                 ; initialize number of shifts

                                ; Read the next word

007553' 201 01 0 00 000200              MOVEI   1,200                   ; get start address
007554' 242 01 0 00 000001              LSH     1,1                     ; position correctly
007555' 260 17 0 00 007437*             GO      LDRAR                   ; load the register
007556' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
007557' 260 17 0 00 007445*             GO      LDCSR                   ; start up the port
007560' 400 01 0 00 000000              SETZ    1,                      ; clear data
007561' 260 17 0 00 007557*             GO      LDCSR                   ; stop the port
007562' 263 17 0 00 000000              RTN                             ; return

                                ; Increment shift count

007563' 201 00 0 00 000004      TS27NN: MOVEI   4                       ; get a 4
007564' 272 00 0 00 014053'             ADDM    TS27NM                  ; add it
007565' 263 17 0 00 000000              RTN                             ; return

007566'                         TS27RS: BLOCK   12                      ; correct EBUF data

                                ; Error messages

007600' 240000  007602'         MA27:   CALL!TXNOT!MA27PN               ; print data pattern ...
007601' 000000000000#                   LAST!CALL!TXALL!TTPNT           ; print test data

007602' 037 00 0 00 013336'     MA27PN: TMSGC   <Data >                 ; get data and
007603' 200 01 0 00 014030'             MOVE    1,T27BUF                ;   print it
007604' 200 00 0 01 000000              MOVE    (1)
007605' 037 13 0 00 000000              PNTHW
007606' 037 00 0 00 013502'             TMSG    < shifted down from PLIN >
007607' 200 00 0 00 014053'             MOVE    TS27NM                  ; get shift count
007610' 037 15 0 00 000000              PNTDEC
007611' 037 00 0 00 013343'             TMSG    <. bits, result in EBUF>
007612' 263 17 0 00 000000              RTN
```

```

                                   ; Microcode - Shift 4 bits

007613' 010001  010000             T27M:   MWORD   <ADDR=100,JMAP,J=101,D=1>                      ; 100
007614' 001000  000040
007615' 010100  000124                     MWORD   <CONT,D=1,SELF,MGC=124>                        ; 101
007616' 001000  003340
007617' 010200  000200                     MWORD   <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>             ; 102
007620' 732002  002340
007621' 010351  000000                     MWORD   <JMAP,J=5100,D=1>                              ; 103
007622' 001000  000040

                                   ;       Routine to read a word from CBUS into the formatter.

007623' 020002  010000                     MWORD   <ADDR=200,JMAP,J=201>                          ; 200
007624' 000000  000040
007625' 020102  030000                     MWORD   <CJP,J=203,D=1,CENA,CCCA>                      ; 201
007626' 001400  000060
007627' 020202  010000                     MWORD   <JMAP,J=201,D=1>                               ; 202
007630' 001000  000040
007631' 020300  000200                     MWORD   <CONT,D=1,SELF,MGC=200>                        ; 203
007632' 001000  003340
007633' 020400  000204                     MWORD   <CONT,D=1,SELF,MGC=204>                        ; 204
007634' 001000  003340
007635' 020500  000004                     MWORD   <CONT,D=1,SELC,MGC=4>                          ; 205
007636' 001000  004340
007637' 020600  000200                     MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>             ; 206
007640' 732001  402340
007641' 020700  000000                     MWORD   <JZ>                                           ; 207
007642' 000000  000000

                                   ;       Initialization

007643' 100010  010020                     MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>            ; 1000
007644' 000000  002040
007645' 100100  000200                     MWORD   <CONT,D=1,SELC,MGC=200>                        ; 1001
007646' 001000  004340
007647' 100200  310017                     MWORD   <LDCT,J=31,SDO,OR,D=2,SKCN,MGC=017>            ; 1002
007650' 732000  240300
007651' 100310  030000                     MWORD   <RPCT,J=1003,SOA,OR,D=5>                       ; 1003
007652' 435000  000220
007653' 100400  002040                     MWORD   <CONT,SOA,OR,D=1,OENA,SELM,MGC=40>             ; 1004
007654' 431000  002340
007655' 100500  000006                     MWORD   <JZ,SDO,D=1,SELP,MGC=6>                        ; 1005
007656' 701000  001000
007657' 000000  000000                     MWORD   <ADDR=0,JZ>                                    ; 0
007660' 000000  000000

                                   ;       Initialization step

007661' 500050  010100                     MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>           ; 5000
007662' 000000  004040
007663' 500100  000020                     MWORD   <JZ,SELM,MGC=20>                               ; 5001
007664' 000000  002000

```

```
                                               ; Subroutine - Passed - Write 1 to bit 12 of CSR register

007665' 510000  000000          MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>                  ; 5100
007666' 742001  000340
007667' 510100  000000          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                       ; 5101
007670' 302001  000740
007671' 510200  260000          MWORD   <LDCT,J=26>                                       ; 5102
007672' 000000  000300
007673' 510351  030000          MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>                ; 5103
007674' 102021  000220
007675' 510400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>             ; 5104
007676' 431020  005340
007677' 510551  070040          MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>            ; 5105
007700' 001400  015060
007701' 510651  050000          MWORD   <JMAP,J=5105,D=1>                                 ; 5106
007702' 001000  000040
007703' 510700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>                ; 5107
007704' 431010  005340
007705' 511051  102004          MWORD   <JMAP,J=5110,D=1,SOA,A=4,OR,OENA,SELE,MGC=4>      ; 5110
007706' 431040  005040
007707' 777777  777777          -1
```

```

                                ;#*******************************************************************
                                ;* Test 30 - Fmtr Down Shift 4 (from PLIN) Test - SELRHTNIB=0
                                ;
                                ; Description:  Verify that the formatter can shift various  data
                                ;               patterns down 4 bits at a  time  while  inserting
                                ;               bits from the PLIN into the  formatter.  The data
                                ;               loaded into the PLIN initially is 360.
                                ;
                                ; Procedure:    KL> Port Clear
                                ;               KL> Set up data transfer for data patterns
                                ;
                                ;               UC> Read a data pattern over the CBUS
                                ;               UC> Load PLIN with 360
                                ;
                                ;               UC> Shift down 4 bits
                                ;               UC> Read formatter contents and write into EBUF
                                ;               KL> Read EBUF and verify the data
                                ;               KL> Start up the ucode to shift again
                                ;
                                ;               Repeat the above shifting then verifying EBUF
                                ;                 contents 10 times, until the data becomes
                                ;                 all zero.
                                ;
                                ;               Do for data patterns:   0's
                                ;                                       1's
                                ;                                       525252,,525252
                                ;                                       000000,,007777
                                ;                                       000000,,007070
                                ;                                       000000,,000707
                                ;
                                ; Failure:      ---
                                ;#*******************************************************************

                                ; Test data

007710' 254 00 0 00 007720'     TSTC30: JRST    TG30                    ; go start test
007711' 100604  000030                  CBUS!NDMP!NDCB!ZCBUS!30         ; test mask
007712' 010127' 013470'                 T30M,,[ASCIZ ^Fmtr Down Shift 4 (from PLIN) Test - SELRHTNIB=0^]
007713' 013263' 012716'                 [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
007714' 000000  010224'                 TSTC31                          ; failure test table
007715' 000000  010542'                 TSTC32                          ;   ...
007716' 000000  011060'                 TSTC33
007717' 777777  777777                  -1

                                ; Start test

007720' 201 00 0 00 000000'     TG30:   MOVEI   Z9                      ; get address of module start
007721' 260 17 0 00 007405*             GO      TRACE                   ; handle trace output
007722' 201 01 0 00 010127'             MOVEI   1,T30M                  ; set up microcode address
007723' 260 17 0 00 007407*             GO      TLOAD                   ; load/verify it
007724' 263 17 0 00 000000              RTN                             ; failed - exit test
```

```

                                        ; Initialization

007725' 400 15 0 00 000000      TL30:   SETZ    ERFLG,                  ; clear error flag
007726' 260 17 0 00 007412*             GO      IPACLR                  ; clear port
007727' 402 00 0 00 007413*             SETZM   TSTSUB                  ; initialize subtest number
                                        MOVSI   [000000,,000000         ; get address of buffer
                                                 777777,,777777         ;   contents
                                                 525252,,525252
                                                 000000,,007777
                                                 000000,,007070
007730' 205 00 0 00 013330'                      000000,,000707]
007731' 541 00 0 00 007430*             HRRI    BUFF                    ; build the BLT pointer
007732' 251 00 0 00 000000#             BLT     BUFF+5                  ; insert data into buffer

                                ; First ensure the data transfer is reset

007733' 201 01 0 00 005000              MOVEI   1,5000                  ; get start address
007734' 242 01 0 00 000001              LSH     1,1                     ; position correctly
007735' 260 17 0 00 007555*             GO      LDRAR                   ; load the register
007736' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
007737' 260 17 0 00 007561*             GO      LDCSR                   ; start up the port
007740' 400 01 0 00 000000              SETZ    1,                      ; clear data
007741' 260 17 0 00 007737*             GO      LDCSR                   ; stop the port

                                ; Now generate CCW list

007742' 200 01 0 00 013210'             MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
007743' 260 17 0 00 007427*             GO      CHINIT                  ; initialize software
007744' 551 01 0 00 007731*             HRRZI   1,BUFF                  ; buffer address
007745' 201 02 0 00 000006              MOVEI   2,6                     ; word count
007746' 400 03 0 00 000000              SETZ    3,                      ; clear zero fill specifier
007747' 260 17 0 00 007433*             GO      GENCCW                  ; generate a CCW list
007750' 201 06 0 00 007773'             MOVEI   6,TS30                  ; get test execute table address

                                ; Start up the data transfer

007751' 201 01 0 00 001000              MOVEI   1,1000                  ; get start address
007752' 242 01 0 00 000001              LSH     1,1                     ; position correctly
007753' 260 17 0 00 007735*             GO      LDRAR                   ; load the register
007754' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
007755' 260 17 0 00 007741*             GO      LDCSR                   ; start up the port
007756' 201 00 0 00 000001              MOVEI   1                       ; set up delay of 1 ms
007757' 260 17 0 00 007443*             GO      ODELAY                  ; wait a bit
007760' 400 01 0 00 000000              SETZ    1,                      ; clear data
007761' 260 17 0 00 007755*             GO      LDCSR                   ; stop the port
```

```

                                       ; Loop on test execute table entries

007762' 260 17 0 00 007446*      TA30:   GO      TEXEC                   ; execute table entry
007763' 254 00 0 00 007772'              JRST    TX30                    ; end of table
007764' 254 00 0 00 007762'              JRST    TA30                    ; keep looping after call
007765' 474 15 0 00 000000               SETO    ERFLG,                  ; error - EBUF data incorrect

                                 ; Handle error printouts and scope looping

007766' 027 00 0 00 010114'              SCOPER  MA30                    ; print error message
007767' 254 00 0 00 007725'              JRST    TL30                    ; loop on error
007770' 254 00 0 00 007772'              JRST    TX30                    ; altmode exit
007771' 322 15 0 00 007762'              JUMPE   ERFLG,TA30              ; do next sstep table entry

                                 ; End of test

007772' 263 17 0 00 000000       TX30:   RTN                             ; return

                                 ; Test Execute Table, as:  (CMD,parameters)

007773' 140000  010047'          TS30:   TTABLE  (TCALL,TS30IN)          ; initialize stuff
007774' 200000  010052'          TS30L:  TTABLE  (TCALLC,TS30NE)         ; set up next data pattern

                                 ; Shifted 4 bits

007775' 140000  010077'          TS30A:  TTABLE  (TCALL,TS30NN)          ; increment shift count
007776' 400000  000100                   TTABLE  (TSSTAR,100)            ; start up port
007777' 240001  010102'                  TTABLE  (TCHECK,1,TS30RS)       ; check if completed
010000' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

                                 ; Shifted 8 bits

010001' 140000  010077'                  TTABLE  (TCALL,TS30NN)          ; increment shift count
010002' 400000  000100                   TTABLE  (TSSTAR,100)            ; start up port
010003' 240001  010103'                  TTABLE  (TCHECK,1,TS30RS+1)     ; check if completed
010004' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

                                 ; Shifted 12 bits

010005' 140000  010077'                  TTABLE  (TCALL,TS30NN)          ; increment shift count
010006' 400000  000100                   TTABLE  (TSSTAR,100)            ; start up port
010007' 240001  010104'                  TTABLE  (TCHECK,1,TS30RS+2)     ; check if completed
010010' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

                                 ; Shifted 16 bits

010011' 140000  010077'                  TTABLE  (TCALL,TS30NN)          ; increment shift count
010012' 400000  000100                   TTABLE  (TSSTAR,100)            ; start up port
010013' 240001  010105'                  TTABLE  (TCHECK,1,TS30RS+3)     ; check if completed
010014' 340000  000000                   TTABLE  (TEXIT)                 ; exit if error yet

```

```

                                 ; Shifted 20 bits

010015' 140000  010077'                   TTABLE  (TCALL,TS30NN)          ; increment shift count
010016' 400000  000100                    TTABLE  (TSSTAR,100)            ; start up port
010017' 240001  010106'                   TTABLE  (TCHECK,1,TS30RS+4)     ; check if completed
010020' 340000  000000                    TTABLE  (TEXIT)                 ; exit if error yet

                                 ; Shifted 24 bits

010021' 140000  010077'                   TTABLE  (TCALL,TS30NN)          ; increment shift count
010022' 400000  000100                    TTABLE  (TSSTAR,100)            ; start up port
010023' 240001  010107'                   TTABLE  (TCHECK,1,TS30RS+5)     ; check if completed
010024' 340000  000000                    TTABLE  (TEXIT)                 ; exit if error yet

                                 ; Shifted 28 bits

010025' 140000  010077'                   TTABLE  (TCALL,TS30NN)          ; increment shift count
010026' 400000  000100                    TTABLE  (TSSTAR,100)            ; start up port
010027' 240001  010110'                   TTABLE  (TCHECK,1,TS30RS+6)     ; check if completed
010030' 340000  000000                    TTABLE  (TEXIT)                 ; exit if error yet

                                 ; Shifted 32 bits

010031' 140000  010077'                   TTABLE  (TCALL,TS30NN)          ; increment shift count
010032' 400000  000100                    TTABLE  (TSSTAR,100)            ; start up port
010033' 240001  010111'                   TTABLE  (TCHECK,1,TS30RS+7)     ; check if completed
010034' 340000  000000                    TTABLE  (TEXIT)                 ; exit if error yet

                                 ; Shifted 36 bits

010035' 140000  010077'                   TTABLE  (TCALL,TS30NN)          ; increment shift count
010036' 400000  000100                    TTABLE  (TSSTAR,100)            ; start up port
010037' 240001  010112'                   TTABLE  (TCHECK,1,TS30RS+10)    ; check if completed
010040' 340000  000000                    TTABLE  (TEXIT)                 ; exit if error yet

                                 ; Shifted 40 bits

010041' 140000  010077'                   TTABLE  (TCALL,TS30NN)          ; increment shift count
010042' 400000  000100                    TTABLE  (TSSTAR,100)            ; start up port
010043' 240001  010113'                   TTABLE  (TCHECK,1,TS30RS+11)    ; check if completed
010044' 340000  000000                    TTABLE  (TEXIT)                 ; exit if error yet
010045' 300000  007774'                   TTABLE  (TJRST,TS30L)           ; loop till done
010046' 000000  000000                    TTABLE  (TLAST)

                                 ; Initialize stuff

010047' 201 00 0 00 000000#      TS30IN:  MOVEI   BUFF-1                  ; initialize buffer pointer
010050' 202 00 0 00 014031'               MOVEM   T30BUF#                 ;  ...
010051' 263 17 0 00 000000                RTN                             ; return
```

```

                                      ; Set up next data pattern

 010052' 350 01 0 00 014031'          TS30NE: AOS     1,T30BUF                 ; point to next data pattern
 010053' 307 01 0 00 000000#                  CAIG    1,BUFF+5                 ; done yet?
 010054' 350 00 0 17 000000                   AOS     (P)                      ; no - set up proper return
 010055' 200 00 0 01 000000                   MOVE    (1)                      ; get data pattern
 010056' 201 02 0 00 010102'                  MOVEI   2,TS30RS                 ; get initial address of EBUF data
 010057' 242 00 0 00 777774           TS30NO: LSH     -4                       ; shift 4 bits
 010060' 303 02 0 00 010103'                  CAILE   2,TS30RS+1               ; if past the first 2 shifts, insert
 010061' 661 00 0 00 000000                   TLO     0                        ;   the PLIN data into left 4 bits
 010062' 202 00 0 02 000000                   MOVEM   (2)                      ; save it
 010063' 350 00 0 00 000002                   AOS     2                        ; point to next address
 010064' 307 02 0 00 010113'                  CAIG    2,TS30RS+11              ; done yet?
 010065' 254 00 0 00 010057'                  JRST    TS30NO                   ; no - loop till done
 010066' 402 00 0 00 014054'                  SETZM   TS30NM#                  ; initialize number of shifts

                                      ; Read the next word

 010067' 201 01 0 00 000200                   MOVEI   1,200                    ; get start address
 010070' 242 01 0 00 000001                   LSH     1,1                      ; position correctly
 010071' 260 17 0 00 007753*                  GO      LDRAR                    ; load the register
 010072' 201 01 0 00 000010                   MOVEI   1,MPRUN                  ; get MPRUN bit
 010073' 260 17 0 00 007761*                  GO      LDCSR                    ; start up the port
 010074' 400 01 0 00 000000                   SETZ    1,                       ; clear data
 010075' 260 17 0 00 010073*                  GO      LDCSR                    ; stop the port
 010076' 263 17 0 00 000000                   RTN                              ; return

                                      ; Increment shift count

 010077' 201 00 0 00 000004           TS30NN: MOVEI   4                        ; get a 4
 010100' 272 00 0 00 014054'                  ADDM    TS30NM                   ; add it
 010101' 263 17 0 00 000000                   RTN                              ; return

 010102'                              TS30RS: BLOCK   12                       ; correct EBUF data

                                      ; Error messages

 010114' 240000  010116'              MA30:   CALL!TXNOT!MA30PN                ; print data pattern ...
 010115' 000000000000#                        LAST!CALL!TXALL!TTPNT            ; print test data

 010116' 037 00 0 00 013336'          MA30PN: TMSGC   <Data >                  ; get data and
 010117' 200 01 0 00 014031'                  MOVE    1,T30BUF                 ;   print it
 010120' 200 00 0 01 000000                   MOVE    (1)
 010121' 037 13 0 00 000000                   PNTHW
 010122' 037 00 0 00 013502'                  TMSG    < shifted down from PLIN >
 010123' 200 00 0 00 014054'                  MOVE    TS30NM                   ; get shift count
 010124' 037 15 0 00 000000                   PNTDEC
 010125' 037 00 0 00 013343'                  TMSG    <. bits, result in EBUF>
 010126' 263 17 0 00 000000                   RTN
```

```

                                        ; Microcode - Shift 4 bits

010127' 010001  010000                  T30M:   MWORD   <ADDR=100,JMAP,J=101,D=1>                   ; 100
010130' 001000  000040
010131' 010100  000124                          MWORD   <CONT,D=1,SELF,MGC=124>                     ; 101
010132' 001000  003340
010133' 010200  000200                          MWORD   <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>          ; 102
010134' 732002  002340
010135' 010351  000000                          MWORD   <JMAP,J=5100,D=1>                           ; 103
010136' 001000  000040

                                        ;       Routine to read a word from CBUS into the formatter.

010137' 020002  010000                          MWORD   <ADDR=200,JMAP,J=201>                       ; 200
010140' 000000  000040
010141' 020102  030000                          MWORD   <CJP,J=203,D=1,CENA,CCCA>                   ; 201
010142' 001400  000060
010143' 020202  010000                          MWORD   <JMAP,J=201,D=1>                            ; 202
010144' 001000  000040
010145' 020300  000200                          MWORD   <CONT,D=1,SELF,MGC=200>                     ; 203
010146' 001000  003340
010147' 020400  000204                          MWORD   <CONT,D=1,SELF,MGC=204>                     ; 204
010150' 001000  003340
010151' 020500  000004                          MWORD   <CONT,D=1,SELC,MGC=4>                       ; 205
010152' 001000  004340
010153' 020600  000200                          MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>          ; 206
010154' 732001  402340
010155' 020700  000000                          MWORD   <JZ>                                        ; 207
010156' 000000  000000

                                        ;       Initialization

010157' 100010  010020                          MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>         ; 1000
010160' 000000  002040
010161' 100100  000200                          MWORD   <CONT,D=1,SELC,MGC=200>                     ; 1001
010162' 001000  004340
010163' 100200  310360                          MWORD   <LDCT,J=31,SDO,OR,D=2,SKCN,MGC=360>         ; 1002
010164' 732000  240300
010165' 100310  030000                          MWORD   <RPCT,J=1003,SOA,OR,D=5>                    ; 1003
010166' 435000  000220
010167' 100400  002040                          MWORD   <CONT,SOA,OR,D=1,OENA,SELM,MGC=40>          ; 1004
010170' 431000  002340
010171' 100500  000006                          MWORD   <JZ,SDO,D=1,SELP,MGC=6>                     ; 1005
010172' 701000  001000
010173' 000000  000000                          MWORD   <ADDR=0,JZ>                                 ; 0
010174' 000000  000000

                                        ;       Initialization step

010175' 500050  010100                          MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>        ; 5000
010176' 000000  004040
010177' 500100  000020                          MWORD   <JZ,SELM,MGC=20>                            ; 5001
010200' 000000  002000

```

```
                                        ; Subroutine - Passed - Write 1 to bit 12 of CSR register

010201' 510000  000000                          MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>                ; 5100
010202' 742001  000340
010203' 510100  000000                          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                     ; 5101
010204' 302001  000740
010205' 510200  260000                          MWORD   <LDCT,J=26>                                     ; 5102
010206' 000000  000300
010207' 510351  030000                          MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>              ; 5103
010210' 102021  000220
010211' 510400  002004                          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>           ; 5104
010212' 431020  005340
010213' 510551  070040                          MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>          ; 5105
010214' 001400  015060
010215' 510651  050000                          MWORD   <JMAP,J=5105,D=1>                               ; 5106
010216' 001000  000040
010217' 510700  000200                          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>              ; 5107
010220' 431010  005340
010221' 511051  102004                          MWORD   <JMAP,J=5110,SOA,A=4,OR,D=1,OENA,SELE,MGC=4>    ; 5110
010222' 431040  005040
010223' 777777  777777                          -1
```

```
                                           ;#******************************************************************
                                           ;* Test 31 - Fmtr Down Shift 4 (from PLIN) Test - SELRHTNIB=1
                                           ;
                                           ; Description:  Verify that the formatter can shift various  data
                                           ;               patterns down 4 bits at a  time  while  inserting
                                           ;               bits from the PLIN into the  formatter.  The data
                                           ;               loaded into the PLIN initially is 017.
                                           ;
                                           ; Procedure:    KL> Port Clear
                                           ;               KL> Set up data transfer for data patterns
                                           ;
                                           ;               UC> Read a data pattern over the CBUS
                                           ;               UC> Load PLIN with 017
                                           ;
                                           ;               UC> Shift down 4 bits
                                           ;               UC> Read formatter contents and write into EBUF
                                           ;               KL> Read EBUF and verify the data
                                           ;               KL> Start up the ucode to shift again
                                           ;
                                           ;               Repeat the above shifting then verifying EBUF
                                           ;                 contents 10 times, until the data becomes
                                           ;                 all zero.
                                           ;
                                           ;               Do for data patterns:   0's
                                           ;                                       1's
                                           ;                                       525252,,525252
                                           ;                                       000000,,007777
                                           ;                                       000000,,007070
                                           ;                                       000000,,000707
                                           ;
                                           ; Failure:      ---
                                           ;#******************************************************************

                                           ; Test data

 010224' 254 00 0 00 010236'               TSTC31: JRST    TG31                            ; go start test
 010225' 100604  000031                            CBUS!NDMP!NDCB!ZCBUS!31                 ; test mask
 010226' 010445' 013507'                           T31M,,[ASCIZ ^Fmtr Down Shift 4 (from PLIN) Test - SELRHTNIB=1^]
 010227' 013263' 012716'                           [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
 010230' 000000  010542'                           TSTC32                                  ; failure test table
 010231' 000000  011060'                           TSTC33                                  ;   ...
 010232' 000000  011344'                           TSTC34
 010233' 000000  011627'                           TSTC35
 010234' 000000  012111'                           TSTC36
 010235' 777777  777777                            -1

                                           ; Start test

 010236' 201 00 0 00 000000'               TG31:   MOVEI   Z9                              ; get address of module start
 010237' 260 17 0 00 007721*                       GO      TRACE                           ; handle trace output
 010240' 201 01 0 00 010445'                       MOVEI   1,T31M                          ; set up microcode address
 010241' 260 17 0 00 007723*                       GO      TLOAD                           ; load/verify it
 010242' 263 17 0 00 000000                        RTN                                     ; failed - exit test
```

```

                                      ; Initialization

010243' 400 15 0 00 000000            TL31:   SETZ    ERFLG,                  ; clear error flag
010244' 260 17 0 00 007726*                   GO      IPACLR                  ; clear port
010245' 402 00 0 00 007727*                   SETZM   TSTSUB                  ; initialize subtest number
                                              MOVSI   [000000,,000000         ; get address of buffer
                                                       777777,,777777         ;   contents
                                                       525252,,525252
                                                       000000,,007777
                                                       000000,,007070
010246' 205 00 0 00 013330'                            000000,,000707]
010247' 541 00 0 00 007744*                   HRRI    BUFF                    ; build the BLT pointer
010250' 251 00 0 00 000000#                   BLT     BUFF+5                  ; insert data into buffer

                                      ; First ensure the data transfer is reset

010251' 201 01 0 00 005000                    MOVEI   1,5000                  ; get start address
010252' 242 01 0 00 000001                    LSH     1,1                     ; position correctly
010253' 260 17 0 00 010071*                   GO      LDRAR                   ; load the register
010254' 201 01 0 00 000010                    MOVEI   1,MPRUN                 ; get MPRUN bit
010255' 260 17 0 00 010075*                   GO      LDCSR                   ; start up the port
010256' 400 01 0 00 000000                    SETZ    1,                      ; clear data
010257' 260 17 0 00 010255*                   GO      LDCSR                   ; stop the port

                                      ; Now generate CCW list

010260' 200 01 0 00 013210'                   MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
010261' 260 17 0 00 007743*                   GO      CHINIT                  ; initialize software
010262' 551 01 0 00 010247*                   HRRZI   1,BUFF                  ; buffer address
010263' 201 02 0 00 000006                    MOVEI   2,6                     ; word count
010264' 400 03 0 00 000000                    SETZ    3,                      ; clear zero fill specifier
010265' 260 17 0 00 007747*                   GO      GENCCW                  ; generate a CCW list
010266' 201 06 0 00 010311'                   MOVEI   6,TS31                  ; get test execute table address

                                      ; Start up the data transfer

010267' 201 01 0 00 001000                    MOVEI   1,1000                  ; get start address
010270' 242 01 0 00 000001                    LSH     1,1                     ; position correctly
010271' 260 17 0 00 010253*                   GO      LDRAR                   ; load the register
010272' 201 01 0 00 000010                    MOVEI   1,MPRUN                 ; get MPRUN bit
010273' 260 17 0 00 010257*                   GO      LDCSR                   ; start up the port
010274' 201 00 0 00 000001                    MOVEI   1                       ; set up delay of 1 ms
010275' 260 17 0 00 007757*                   GO      ODELAY                  ; wait a bit
010276' 400 01 0 00 000000                    SETZ    1,                      ; clear data
010277' 260 17 0 00 010273*                   GO      LDCSR                   ; stop the port
```

```

                                        ; Loop on test execute table entries

010300' 260 17 0 00 007762*     TA31:   GO      TEXEC                   ; execute table entry
010301' 254 00 0 00 010310'             JRST    TX31                    ; end of table
010302' 254 00 0 00 010300'             JRST    TA31                    ; keep looping after call
010303' 474 15 0 00 000000              SETO    ERFLG,                  ; error - EBUF data incorrect

                                        ; Handle error printouts and scope looping

010304' 027 00 0 00 010432'             SCOPER  MA31                    ; print error message
010305' 254 00 0 00 010243'             JRST    TL31                    ; loop on error
010306' 254 00 0 00 010310'             JRST    TX31                    ; altmode exit
010307' 322 15 0 00 010300'             JUMPE   ERFLG,TA31              ; do next sstep table entry

                                        ; End of test

010310' 263 17 0 00 000000      TX31:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

010311' 140000  010365'         TS31:   TTABLE  (TCALL,TS31IN)          ; initialize stuff
010312' 200000  010370'         TS31L:  TTABLE  (TCALLC,TS31NE)         ; set up next data pattern

                                        ; Shifted 4 bits

010313' 140000  010415'         TS31A:  TTABLE  (TCALL,TS31NN)          ; increment shift count
010314' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
010315' 240001  010420'                 TTABLE  (TCHECK,1,TS31RS)       ; check if completed
010316' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 8 bits

010317' 140000  010415'                 TTABLE  (TCALL,TS31NN)          ; increment shift count
010320' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
010321' 240001  010421'                 TTABLE  (TCHECK,1,TS31RS+1)     ; check if completed
010322' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 12 bits

010323' 140000  010415'                 TTABLE  (TCALL,TS31NN)          ; increment shift count
010324' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
010325' 240001  010422'                 TTABLE  (TCHECK,1,TS31RS+2)     ; check if completed
010326' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 16 bits

010327' 140000  010415'                 TTABLE  (TCALL,TS31NN)          ; increment shift count
010330' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
010331' 240001  010423'                 TTABLE  (TCHECK,1,TS31RS+3)     ; check if completed
010332' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

```

```

                                         ; Shifted 20 bits

010333' 140000  010415'                          TTABLE   (TCALL,TS31NN)          ; increment shift count
010334' 400000  000100                           TTABLE   (TSSTAR,100)            ; start up port
010335' 240001  010424'                          TTABLE   (TCHECK,1,TS31RS+4)     ; check if completed
010336' 340000  000000                           TTABLE   (TEXIT)                 ; exit if error yet

                                         ; Shifted 24 bits

010337' 140000  010415'                          TTABLE   (TCALL,TS31NN)          ; increment shift count
010340' 400000  000100                           TTABLE   (TSSTAR,100)            ; start up port
010341' 240001  010425'                          TTABLE   (TCHECK,1,TS31RS+5)     ; check if completed
010342' 340000  000000                           TTABLE   (TEXIT)                 ; exit if error yet

                                         ; Shifted 28 bits

010343' 140000  010415'                          TTABLE   (TCALL,TS31NN)          ; increment shift count
010344' 400000  000100                           TTABLE   (TSSTAR,100)            ; start up port
010345' 240001  010426'                          TTABLE   (TCHECK,1,TS31RS+6)     ; check if completed
010346' 340000  000000                           TTABLE   (TEXIT)                 ; exit if error yet

                                         ; Shifted 32 bits

010347' 140000  010415'                          TTABLE   (TCALL,TS31NN)          ; increment shift count
010350' 400000  000100                           TTABLE   (TSSTAR,100)            ; start up port
010351' 240001  010427'                          TTABLE   (TCHECK,1,TS31RS+7)     ; check if completed
010352' 340000  000000                           TTABLE   (TEXIT)                 ; exit if error yet

                                         ; Shifted 36 bits

010353' 140000  010415'                          TTABLE   (TCALL,TS31NN)          ; increment shift count
010354' 400000  000100                           TTABLE   (TSSTAR,100)            ; start up port
010355' 240001  010430'                          TTABLE   (TCHECK,1,TS31RS+10)    ; check if completed
010356' 340000  000000                           TTABLE   (TEXIT)                 ; exit if error yet

                                         ; Shifted 40 bits

010357' 140000  010415'                          TTABLE   (TCALL,TS31NN)          ; increment shift count
010360' 400000  000100                           TTABLE   (TSSTAR,100)            ; start up port
010361' 240001  010431'                          TTABLE   (TCHECK,1,TS31RS+11)    ; check if completed
010362' 340000  000000                           TTABLE   (TEXIT)                 ; exit if error yet
010363' 300000  010312'                          TTABLE   (TJRST,TS31L)           ; loop till done
010364' 000000  000000                           TTABLE   (TLAST)

                                         ; Initialize stuff

010365' 201 00 0 00 000000#              TS31IN: MOVEI    BUFF-1                  ; initialize buffer pointer
010366' 202 00 0 00 014032'                      MOVEM    T31BUF#                 ;  ...
010367' 263 17 0 00 000000                       RTN                              ; return
```

```

                                        ; Set up next data pattern

010370' 350 01 0 00 014032'     TS31NE: AOS     1,T31BUF                ; point to next data pattern
010371' 307 01 0 00 000000#             CAIG    1,BUFF+5                ; done yet?
010372' 350 00 0 17 000000              AOS     (P)                     ; no - set up proper return
010373' 200 00 0 01 000000              MOVE    (1)                     ; get data pattern
010374' 201 02 0 00 010420'             MOVEI   2,TS31RS                ; get initial address of EBUF data
010375' 242 00 0 00 777774      TS31NO: LSH     -4                      ; shift 4 bits
010376' 303 02 0 00 010421'             CAILE   2,TS31RS+1              ; if past the first 2 shifts, insert
010377' 661 00 0 00 000000              TLO     0                       ;   the PLIN data into left 4 bits
010400' 202 00 0 02 000000              MOVEM   (2)                     ; save it
010401' 350 00 0 00 000002              AOS     2                       ; point to next address
010402' 307 02 0 00 010431'             CAIG    2,TS31RS+11             ; done yet?
010403' 254 00 0 00 010375'             JRST    TS31NO                  ; no - loop till done
010404' 402 00 0 00 014055'             SETZM   TS31NM#                 ; initialize number of shifts

                                        ; Read the next word

010405' 201 01 0 00 000200              MOVEI   1,200                   ; get start address
010406' 242 01 0 00 000001              LSH     1,1                     ; position correctly
010407' 260 17 0 00 010271*             GO      LDRAR                   ; load the register
010410' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
010411' 260 17 0 00 010277*             GO      LDCSR                   ; start up the port
010412' 400 01 0 00 000000              SETZ    1,                      ; clear data
010413' 260 17 0 00 010411*             GO      LDCSR                   ; stop the port
010414' 263 17 0 00 000000              RTN                             ; return

                                        ; Increment shift count

010415' 201 00 0 00 000004      TS31NN: MOVEI   4                       ; get a 4
010416' 272 00 0 00 014055'             ADDM    TS31NM                  ; add it
010417' 263 17 0 00 000000              RTN                             ; return

010420'                         TS31RS: BLOCK   12                      ; correct EBUF data

                                        ; Error messages

010432' 240000  010434'         MA31:   CALL!TXNOT!MA31PN               ; print data pattern ...
010433' 000000000000#                   LAST!CALL!TXALL!TTPNT           ; print test data

010434' 037 00 0 00 013336'     MA31PN: TMSGC   <Data >                 ; get data and
010435' 200 01 0 00 014032'             MOVE    1,T31BUF                ;   print it
010436' 200 00 0 01 000000              MOVE    (1)
010437' 037 13 0 00 000000              PNTHW
010440' 037 00 0 00 013502'             TMSG    < shifted down from PLIN >
010441' 200 00 0 00 014055'             MOVE    TS31NM                  ; get shift count
010442' 037 15 0 00 000000              PNTDEC
010443' 037 00 0 00 013343'             TMSG    <. bits, result in EBUF>
010444' 263 17 0 00 000000              RTN
```

```

                                      ; Microcode - Shift 4 bits

010445' 010001  010000                T31M:   MWORD   <ADDR=100,JMAP,J=101,D=1>                       ; 100
010446' 001000  000040
010447' 010100  000126                        MWORD   <CONT,D=1,SELF,MGC=126>                         ; 101
010450' 001000  003340
010451' 010200  000200                        MWORD   <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>              ; 102
010452' 732002  002340
010453' 010351  000000                        MWORD   <JMAP,J=5100,D=1>                               ; 103
010454' 001000  000040

                                      ;       Routine to read a word from CBUS into the formatter.

010455' 020002  010000                        MWORD   <ADDR=200,JMAP,J=201>                           ; 200
010456' 000000  000040
010457' 020102  030000                        MWORD   <CJP,J=203,D=1,CENA,CCCA>                       ; 201
010460' 001400  000060
010461' 020202  010000                        MWORD   <JMAP,J=201,D=1>                                ; 202
010462' 001000  000040
010463' 020300  000200                        MWORD   <CONT,D=1,SELF,MGC=200>                         ; 203
010464' 001000  003340
010465' 020400  000204                        MWORD   <CONT,D=1,SELF,MGC=204>                         ; 204
010466' 001000  003340
010467' 020500  000004                        MWORD   <CONT,D=1,SELC,MGC=4>                           ; 205
010470' 001000  004340
010471' 020600  000200                        MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>              ; 206
010472' 732001  402340
010473' 020700  000000                        MWORD   <JZ>                                            ; 207
010474' 000000  000000

                                      ;       Initialization

010475' 100010  010020                        MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>             ; 1000
010476' 000000  002040
010477' 100100  000200                        MWORD   <CONT,D=1,SELC,MGC=200>                         ; 1001
010500' 001000  004340
010501' 100200  310017                        MWORD   <LDCT,J=31,SDO,OR,D=2,SKCN,MGC=017>             ; 1002
010502' 732000  240300
010503' 100310  030000                        MWORD   <RPCT,J=1003,SOA,OR,D=5>                        ; 1003
010504' 435000  000222
010505' 100400  002040                        MWORD   <CONT,SOA,OR,D=1,OENA,SELM,MGC=40>              ; 1004
010506' 431000  002340
010507' 100500  000006                        MWORD   <JZ,SDO,D=1,SELP,MGC=6>                         ; 1005
010510' 701000  001000
010511' 000000  000000                        MWORD   <ADDR=0,JZ>                                     ; 0
010512' 000000  000000

                                      ;       Initialization step

010513' 500050  010100                        MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>            ; 5000
010514' 000000  004040
010515' 500100  000020                        MWORD   <JZ,SELM,MGC=20>                                ; 5001
010516' 000000  002000

```

```
                                      ; Subroutine - Passed - Write 1 to bit 12 of CSR register

010517' 510000  000000                     MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>                 ; 5100
010520' 742001  000340
010521' 510100  000000                     MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                      ; 5101
010522' 302001  000740
010523' 510200  260000                     MWORD   <LDCT,J=26>                                      ; 5102
010524' 000000  000300
010525' 510351  030000                     MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>               ; 5103
010526' 102021  000220
010527' 510400  002004                     MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>            ; 5104
010530' 431020  005340
010531' 510551  070040                     MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>           ; 5105
010532' 001400  015060
010533' 510651  050000                     MWORD   <JMAP,J=5105,D=1>                                ; 5106
010534' 001000  000040
010535' 510700  000200                     MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>               ; 5107
010536' 431010  005340
010537' 511051  102004                     MWORD   <JMAP,J=5110,SOA,A=4,OR,D=1,OENA,SELE,MGC=4>     ; 5110
010540' 431040  005040
010541' 777777  777777                     -1
```

```

                                ;#*******************************************************************
                                ;* Test 32 - Fmtr Down Shift 4 (from PLIN) Test - SELRHTNIB=1
                                ;
                                ; Description:  Verify that the formatter can shift various  data
                                ;               patterns  down  4  bits at a time while inserting
                                ;               bits from the PLIN into the formatter.  The  data
                                ;               loaded into the PLIN initially is 360.
                                ;
                                ; Procedure:    KL> Port Clear
                                ;               KL> Set up data transfer for data patterns
                                ;
                                ;               UC> Read a data pattern over the CBUS
                                ;               UC> Load PLIN with 360
                                ;
                                ;               UC> Shift down 4 bits
                                ;               UC> Read formatter contents and write into EBUF
                                ;               KL> Read EBUF and verify the data
                                ;               KL> Start up the ucode to shift again
                                ;
                                ;               Repeat the above shifting then verifying EBUF
                                ;                 contents 10 times, until the data becomes
                                ;                 all zero.
                                ;
                                ;               Do for data patterns:   0's
                                ;                                       1's
                                ;                                       525252,,525252
                                ;                                       000000,,007777
                                ;                                       000000,,007070
                                ;                                       000000,,000707
                                ;
                                ; Failure:      ---
                                ;#*******************************************************************

                                ; Test data

010542' 254 00 0 00 010554'     TSTC32: JRST    TG32                    ; go start test
010543' 100604  000032                  CBUS!NDMP!NDCB!ZCBUS!32         ; test mask
010544' 010763' 013507'                 T32M,,[ASCIZ ^Fmtr Down Shift 4 (from PLIN) Test - SELRHTNIB=1^]
010545' 013263' 012716'                 [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
010546' 000000  011060'                 TSTC33                          ; failure test table
010547' 000000  011344'                 TSTC34                          ;   ...
010550' 000000  011627'                 TSTC35
010551' 000000  012111'                 TSTC36
010552' 000000  012372'                 TSTC37
010553' 777777  777777                  -1

                                ; Start test

010554' 201 00 0 00 000000'     TG32:   MOVEI   Z9                      ; get address of module start
010555' 260 17 0 00 010237*             GO      TRACE                   ; handle trace output
010556' 201 01 0 00 010763'             MOVEI   1,T32M                  ; set up microcode address
010557' 260 17 0 00 010241*             GO      TLOAD                   ; load/verify it
010560' 263 17 0 00 000000              RTN                             ; failed - exit test
```

```
                                      ; Initialization

010561' 400 15 0 00 000000      TL32:   SETZ    ERFLG,                  ; clear error flag
010562' 260 17 0 00 010244*             GO      IPACLR                  ; clear port
010563' 402 00 0 00 010245*             SETZM   TSTSUB                  ; initialize subtest number
                                        MOVSI   [000000,,000000         ; get address of buffer
                                                 777777,,777777         ;   contents
                                                 525252,,525252
                                                 000000,,007777
                                                 000000,,007070
010564' 205 00 0 00 013330'                      000000,,000707]
010565' 541 00 0 00 010262*             HRRI    BUFF                    ; build the BLT pointer
010566' 251 00 0 00 000000#             BLT     BUFF+5                  ; insert data into buffer

                                ; First ensure the data transfer is reset

010567' 201 01 0 00 005000              MOVEI   1,5000                  ; get start address
010570' 242 01 0 00 000001              LSH     1,1                     ; position correctly
010571' 260 17 0 00 010407*             GO      LDRAR                   ; load the register
010572' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
010573' 260 17 0 00 010413*             GO      LDCSR                   ; start up the port
010574' 400 01 0 00 000000              SETZ    1,                      ; clear data
010575' 260 17 0 00 010573*             GO      LDCSR                   ; stop the port

                                ; Now generate CCW list

010576' 200 01 0 00 013210'             MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
010577' 260 17 0 00 010261*             GO      CHINIT                  ; initialize software
010600' 551 01 0 00 010565*             HRRZI   1,BUFF                  ; buffer address
010601' 201 02 0 00 000006              MOVEI   2,6                     ; word count
010602' 400 03 0 00 000000              SETZ    3,                      ; clear zero fill specifier
010603' 260 17 0 00 010265*             GO      GENCCW                  ; generate a CCW list
010604' 201 06 0 00 010627'             MOVEI   6,TS32                  ; get test execute table address

                                ; Start up the data transfer

010605' 201 01 0 00 001000              MOVEI   1,1000                  ; get start address
010606' 242 01 0 00 000001              LSH     1,1                     ; position correctly
010607' 260 17 0 00 010571*             GO      LDRAR                   ; load the register
010610' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
010611' 260 17 0 00 010575*             GO      LDCSR                   ; start up the port
010612' 201 00 0 00 000001              MOVEI   1                       ; set up delay of 1 ms
010613' 260 17 0 00 010275*             GO      ODELAY                  ; wait a bit
010614' 400 01 0 00 000000              SETZ    1,                      ; clear data
010615' 260 17 0 00 010611*             GO      LDCSR                   ; stop the port

                                ; Loop on test execute table entries

010616' 260 17 0 00 010300*     TA32:   GO      TEXEC                   ; execute table entry
010617' 254 00 0 00 010626'             JRST    TX32                    ; end of table
010620' 254 00 0 00 010616'             JRST    TA32                    ; keep looping after call
010621' 474 15 0 00 000000              SETO    ERFLG,                  ; error - EBUF data incorrect
```

```

                                        ; Handle error printouts and scope looping

010622' 027 00 0 00 010750'                     SCOPER  MA32                    ; print error message
010623' 254 00 0 00 010561'                     JRST    TL32                    ; loop on error
010624' 254 00 0 00 010626'                     JRST    TX32                    ; altmode exit
010625' 322 15 0 00 010616'                     JUMPE   ERFLG,TA32              ; do next sstep table entry

                                        ; End of test

010626' 263 17 0 00 000000              TX32:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

010627' 140000  010703'                 TS32:   TTABLE  (TCALL,TS32IN)          ; initialize stuff
010630' 200000  010706'                 TS32L:  TTABLE  (TCALLC,TS32NE)         ; set up next data pattern

                                        ; Shifted 4 bits

010631' 140000  010733'                 TS32A:  TTABLE  (TCALL,TS32NN)          ; increment shift count
010632' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
010633' 240001  010736'                         TTABLE  (TCHECK,1,TS32RS)       ; check if completed
010634' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 8 bits

010635' 140000  010733'                         TTABLE  (TCALL,TS32NN)          ; increment shift count
010636' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
010637' 240001  010737'                         TTABLE  (TCHECK,1,TS32RS+1)     ; check if completed
010640' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 12 bits

010641' 140000  010733'                         TTABLE  (TCALL,TS32NN)          ; increment shift count
010642' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
010643' 240001  010740'                         TTABLE  (TCHECK,1,TS32RS+2)     ; check if completed
010644' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 16 bits

010645' 140000  010733'                         TTABLE  (TCALL,TS32NN)          ; increment shift count
010646' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
010647' 240001  010741'                         TTABLE  (TCHECK,1,TS32RS+3)     ; check if completed
010650' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 20 bits

010651' 140000  010733'                         TTABLE  (TCALL,TS32NN)          ; increment shift count
010652' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
010653' 240001  010742'                         TTABLE  (TCHECK,1,TS32RS+4)     ; check if completed
010654' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

```

```

                                        ; Shifted 24 bits

010655' 140000  010733'                         TTABLE  (TCALL,TS32NN)          ; increment shift count
010656' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
010657' 240001  010743'                         TTABLE  (TCHECK,1,TS32RS+5)     ; check if completed
010660' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 28 bits

010661' 140000  010733'                         TTABLE  (TCALL,TS32NN)          ; increment shift count
010662' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
010663' 240001  010744'                         TTABLE  (TCHECK,1,TS32RS+6)     ; check if completed
010664' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 32 bits

010665' 140000  010733'                         TTABLE  (TCALL,TS32NN)          ; increment shift count
010666' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
010667' 240001  010745'                         TTABLE  (TCHECK,1,TS32RS+7)     ; check if completed
010670' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 36 bits

010671' 140000  010733'                         TTABLE  (TCALL,TS32NN)          ; increment shift count
010672' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
010673' 240001  010746'                         TTABLE  (TCHECK,1,TS32RS+10)    ; check if completed
010674' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 40 bits

010675' 140000  010733'                         TTABLE  (TCALL,TS32NN)          ; increment shift count
010676' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
010677' 240001  010747'                         TTABLE  (TCHECK,1,TS32RS+11)    ; check if completed
010700' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet
010701' 300000  010630'                         TTABLE  (TJRST,TS32L)           ; loop till done
010702' 000000  000000                          TTABLE  (TLAST)

                                        ; Initialize stuff

010703' 201 00 0 00 000000#             TS32IN: MOVEI   BUFF-1                  ; initialize buffer pointer
010704' 202 00 0 00 014033'                     MOVEM   T32BUF#                 ;  ...
010705' 263 17 0 00 000000                      RTN                             ; return
```

```

                                   ; Set up next data pattern

010706' 350 01 0 00 014033'        TS32NE: AOS     1,T32BUF                ; point to next data pattern
010707' 307 01 0 00 000000#                CAIG    1,BUFF+5                ; done yet?
010710' 350 00 0 17 000000                 AOS     (P)                     ; no - set up proper return
010711' 200 00 0 01 000000                 MOVE    (1)                     ; get data pattern
010712' 201 02 0 00 010736'                MOVEI   2,TS32RS                ; get initial address of EBUF data
010713' 242 00 0 00 777774         TS32NO: LSH     -4                      ; shift 4 bits
010714' 303 02 0 00 010737'                CAILE   2,TS32RS+1              ; if past the first 2 shifts, insert
010715' 661 00 0 00 740000                 TLO     740000                  ;   the PLIN data into left 4 bits
010716' 202 00 0 02 000000                 MOVEM   (2)                     ; save it
010717' 350 00 0 00 000002                 AOS     2                       ; point to next address
010720' 307 02 0 00 010747'                CAIG    2,TS32RS+11             ; done yet?
010721' 254 00 0 00 010713'                JRST    TS32NO                  ; no - loop till done
010722' 402 00 0 00 014056'                SETZM   TS32NM#                 ; initialize number of shifts

                                   ; Read the next word

010723' 201 01 0 00 000200                 MOVEI   1,200                   ; get start address
010724' 242 01 0 00 000001                 LSH     1,1                     ; position correctly
010725' 260 17 0 00 010607*                GO      LDRAR                   ; load the register
010726' 201 01 0 00 000010                 MOVEI   1,MPRUN                 ; get MPRUN bit
010727' 260 17 0 00 010615*                GO      LDCSR                   ; start up the port
010730' 400 01 0 00 000000                 SETZ    1,                      ; clear data
010731' 260 17 0 00 010727*                GO      LDCSR                   ; stop the port
010732' 263 17 0 00 000000                 RTN                             ; return

                                   ; Increment shift count

010733' 201 00 0 00 000004         TS32NN: MOVEI   4                       ; get a 4
010734' 272 00 0 00 014056'                ADDM    TS32NM                  ; add it
010735' 263 17 0 00 000000                 RTN                             ; return

010736'                            TS32RS: BLOCK   12                      ; correct EBUF data

                                   ; Error messages

010750' 240000  010752'            MA32:   CALL!TXNOT!MA32PN               ; print data pattern ...
010751' 000000000000#                      LAST!CALL!TXALL!TTPNT           ; print test data

010752' 037 00 0 00 013336'        MA32PN: TMSGC   <Data >                 ; get data and
010753' 200 01 0 00 014033'                MOVE    1,T32BUF                ;   print it
010754' 200 00 0 01 000000                 MOVE    (1)
010755' 037 13 0 00 000000                 PNTHW
010756' 037 00 0 00 013502'                TMSG    < shifted down from PLIN >
010757' 200 00 0 00 014056'                MOVE    TS32NM                  ; get shift count
010760' 037 15 0 00 000000                 PNTDEC
010761' 037 00 0 00 013343'                TMSG    <. bits, result in EBUF>
010762' 263 17 0 00 000000                 RTN
```

```

                                       ; Microcode - Shift 4 bits

010763' 010001  010000                 T32M:  MWORD   <ADDR=100,JMAP,J=101,D=1>                  ; 100
010764' 001000  000040
010765' 010100  000126                        MWORD   <CONT,D=1,SELF,MGC=126>                    ; 101
010766' 001000  003340
010767' 010200  000200                        MWORD   <CONT,SD0,OR,B=4,D=2,SELM,MGC=200>         ; 102
010770' 732002  002340
010771' 010351  000000                        MWORD   <JMAP,J=5100,D=1>                          ; 103
010772' 001000  000040

                                       ;      Routine to read a word from CBUS into the formatter.

010773' 020002  010000                        MWORD   <ADDR=200,JMAP,J=201>                      ; 200
010774' 000000  000040
010775' 020102  030000                        MWORD   <CJP,J=203,D=1,CENA,CCCA>                  ; 201
010776' 001400  000060
010777' 020202  010000                        MWORD   <JMAP,J=201,D=1>                           ; 202
011000' 001000  000040
011001' 020300  000200                        MWORD   <CONT,D=1,SELF,MGC=200>                    ; 203
011002' 001000  003340
011003' 020400  000204                        MWORD   <CONT,D=1,SELF,MGC=204>                    ; 204
011004' 001000  003340
011005' 020500  000004                        MWORD   <CONT,D=1,SELC,MGC=4>                      ; 205
011006' 001000  004340
011007' 020600  000200                        MWORD   <CONT,SD0,OR,B=3,D=2,SELM,MGC=200>         ; 206
011010' 732001  402340
011011' 020700  000000                        MWORD   <JZ>                                       ; 207
011012' 000000  000000

                                       ;      Initialization

011013' 100010  010020                        MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>        ; 1000
011014' 000000  002040
011015' 100100  000200                        MWORD   <CONT,D=1,SELC,MGC=200>                    ; 1001
011016' 001000  004340
011017' 100200  310360                        MWORD   <LDCT,J=31,SD0,OR,D=2,SKCN,MGC=360>        ; 1002
011020' 732000  240300
011021' 100310  030000                        MWORD   <RPCT,J=1003,S0A,OR,D=5>                   ; 1003
011022' 435000  000220
011023' 100400  002040                        MWORD   <CONT,S0A,OR,D=1,OENA,SELM,MGC=40>         ; 1004
011024' 431000  002340
011025' 100500  000006                        MWORD   <JZ,SD0,D=1,SELP,MGC=6>                    ; 1005
011026' 701000  001000
011027' 000000  000000                        MWORD   <ADDR=0,JZ>                                ; 0
011030' 000000  000000

                                       ;      Initialization step

011031' 500050  010100                        MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>       ; 5000
011032' 000000  004040
011033' 500100  000020                        MWORD   <JZ,SELM,MGC=20>                           ; 5001
011034' 000000  002000

```

```
 7393                                            ; Subroutine - Passed - Write 1 to bit 12 of CSR register
 7394
 7395   011035' 510000  000000                          MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>                ; 5100
 7396   011036' 742001  000340
 7397   011037' 510100  000000                          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                     ; 5101
 7398   011040' 302001  000740
 7399   011041' 510200  260000                          MWORD   <LDCT,J=26>                                     ; 5102
 7400   011042' 000000  000300
 7401   011043' 510351  030000                          MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>              ; 5103
 7402   011044' 102021  000220
 7403   011045' 510400  002004                          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>           ; 5104
 7404   011046' 431020  005340
 7405   011047' 510551  070040                          MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>          ; 5105
 7406   011050' 001400  015060
 7407   011051' 510651  050000                          MWORD   <JMAP,J=5105,D=1>                               ; 5106
 7408   011052' 001000  000040
 7409   011053' 510700  000200                          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>              ; 5107
 7410   011054' 431010  005340
 7411   011055' 511051  102004                          MWORD   <JMAP,J=5110,SOA,A=4,OR,D=1,OENA,SELE,MGC=4>    ; 5110
 7412   011056' 431040  005040
 7413   011057' 777777  777777                          -1
```

```

                                      ;#*****************************************************************
                                      ;* Test 33 - Fmtr Down Shift 8 (from PLIN) Test - SELRHTNIB=0
                                      ;
                                      ; Description:  Verify that the formatter can shift various  data
                                      ;               patterns down 8 bits at a  time  while  inserting
                                      ;               bits from the PLIN into the  formatter.  The data
                                      ;               loaded into the PLIN initially is 017.
                                      ;
                                      ; Procedure:    KL> Port Clear
                                      ;               KL> Set up data transfer for data patterns
                                      ;
                                      ;               UC> Read a data pattern over the CBUS
                                      ;               UC> Load PLIN with 017
                                      ;
                                      ;               UC> Shift down 8 bits
                                      ;               UC> Read formatter contents and write into EBUF
                                      ;               KL> Read EBUF and verify the data
                                      ;               KL> Start up the ucode to shift again
                                      ;
                                      ;               Repeat the above shifting then verifying EBUF
                                      ;                 contents 5 times, until the data becomes
                                      ;                 all zero.
                                      ;
                                      ;               Do for data patterns:   0's
                                      ;                                       1's
                                      ;                                       525252,,525252
                                      ;                                       000000,,007777
                                      ;                                       000000,,007070
                                      ;                                       000000,,000707
                                      ;
                                      ; Failure:      ---
                                      ;#*****************************************************************

                                      ; Test data

011060' 254 00 0 00 011071'          TSTC33: JRST    TG33                    ; go start test
011061' 100604  000033                       CBUS!NDMP!NDCB!ZCBUS!33         ; test mask
011062' 011247' 013521'                      T33M,,[ASCIZ ^Fmtr Down Shift 8 (from PLIN) Test - SELRHTNIB=0^]
011063' 013263' 012716'                      [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
011064' 000000  011344'                      TSTC34                          ; failure test table
011065' 000000  011627'                      TSTC35                          ;   ...
011066' 000000  012111'                      TSTC36
011067' 000000  012372'                      TSTC37
011070' 777777  777777                       -1

                                      ; Start test

011071' 201 00 0 00 000000'          TG33:   MOVEI   Z9                      ; get address of module start
011072' 260 17 0 00 010555*                  GO      TRACE                   ; handle trace output
011073' 201 01 0 00 011247'                  MOVEI   1,T33M                  ; set up microcode address
011074' 260 17 0 00 010557*                  GO      TLOAD                   ; load/verify it
011075' 263 17 0 00 000000                   RTN                             ; failed - exit test
```

```

                                ; Initialization

011076' 400 15 0 00 000000      TL33:   SETZ    ERFLG,                  ; clear error flag
011077' 260 17 0 00 010562*             GO      IPACLR                  ; clear port
011100' 402 00 0 00 010563*             SETZM   TSTSUB                  ; initialize subtest number
                                        MOVSI   [000000,,000000         ; get address of buffer
                                                 777777,,777777         ;   contents
                                                 525252,,525252
                                                 000000,,007777
                                                 000000,,007070
011101' 205 00 0 00 013330'                      000000,,000707]
011102' 541 00 0 00 010600*             HRRI    BUFF                    ; build the BLT pointer
011103' 251 00 0 00 000000#             BLT     BUFF+5                  ; insert data into buffer

                                ; First ensure the data transfer is reset

011104' 201 01 0 00 005000              MOVEI   1,5000                  ; get start address
011105' 242 01 0 00 000001              LSH     1,1                     ; position correctly
011106' 260 17 0 00 010725*             GO      LDRAR                   ; load the register
011107' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
011110' 260 17 0 00 010731*             GO      LDCSR                   ; start up the port
011111' 400 01 0 00 000000              SETZ    1,                      ; clear data
011112' 260 17 0 00 011110*             GO      LDCSR                   ; stop the port

                                ; Now generate CCW list

011113' 200 01 0 00 013210'             MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
011114' 260 17 0 00 010577*             GO      CHINIT                  ; initialize software
011115' 551 01 0 00 011102*             HRRZI   1,BUFF                  ; buffer address
011116' 201 02 0 00 000006              MOVEI   2,6                     ; word count
011117' 400 03 0 00 000000              SETZ    3,                      ; clear zero fill specifier
011120' 260 17 0 00 010603*             GO      GENCCW                  ; generate a CCW list
011121' 201 06 0 00 011144'             MOVEI   6,TS33                  ; get test execute table address

                                ; Start up the data transfer

011122' 201 01 0 00 001000              MOVEI   1,1000                  ; get start address
011123' 242 01 0 00 000001              LSH     1,1                     ; position correctly
011124' 260 17 0 00 011106*             GO      LDRAR                   ; load the register
011125' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
011126' 260 17 0 00 011112*             GO      LDCSR                   ; start up the port
011127' 201 00 0 00 000001              MOVEI   1                       ; set up delay of 1 ms
011130' 260 17 0 00 010613*             GO      ODELAY                  ; wait a bit
011131' 400 01 0 00 000000              SETZ    1,                      ; clear data
011132' 260 17 0 00 011126*             GO      LDCSR                   ; stop the port
```

```

                                      ; Loop on test execute table entries

011133' 260 17 0 00 010616*           TA33:   GO      TEXEC                           ; execute table entry
011134' 254 00 0 00 011143'                   JRST    TX33                            ; end of table
011135' 254 00 0 00 011133'                   JRST    TA33                            ; keep looping after call
011136' 474 15 0 00 000000                    SETO    ERFLG,                          ; error - EBUF data incorrect

                                      ; Handle error printouts and scope looping

011137' 027 00 0 00 011234'                   SCOPER  MA33                            ; print error message
011140' 254 00 0 00 011076'                   JRST    TL33                            ; loop on error
011141' 254 00 0 00 011143'                   JRST    TX33                            ; altmode exit
011142' 322 15 0 00 011133'                   JUMPE   ERFLG,TA33                      ; do next sstep table entry

                                      ; End of test

011143' 263 17 0 00 000000            TX33:   RTN                                     ; return

                                      ; Test Execute Table, as:  (CMD,parameters)

011144' 140000  011174'               TS33:   TTABLE  (TCALL,TS33IN)                  ; initialize stuff
011145' 200000  011177'               TS33L:  TTABLE  (TCALLC,TS33NE)                 ; set up next data pattern

                                      ; Shifted 8 bits

011146' 140000  011224'               TS33A:  TTABLE  (TCALL,TS33NN)                  ; increment shift count
011147' 400000  000100                        TTABLE  (TSSTAR,100)                    ; start up port
011150' 240001  011227'                       TTABLE  (TCHECK,1,TS33RS)               ; check if completed
011151' 340000  000000                        TTABLE  (TEXIT)                         ; exit if error yet

                                      ; Shifted 16 bits

011152' 140000  011224'                       TTABLE  (TCALL,TS33NN)                  ; increment shift count
011153' 400000  000100                        TTABLE  (TSSTAR,100)                    ; start up port
011154' 240001  011230'                       TTABLE  (TCHECK,1,TS33RS+1)             ; check if completed
011155' 340000  000000                        TTABLE  (TEXIT)                         ; exit if error yet

                                      ; Shifted 24 bits

011156' 140000  011224'                       TTABLE  (TCALL,TS33NN)                  ; increment shift count
011157' 400000  000100                        TTABLE  (TSSTAR,100)                    ; start up port
011160' 240001  011231'                       TTABLE  (TCHECK,1,TS33RS+2)             ; check if completed
011161' 340000  000000                        TTABLE  (TEXIT)                         ; exit if error yet

                                      ; Shifted 32 bits

011162' 140000  011224'                       TTABLE  (TCALL,TS33NN)                  ; increment shift count
011163' 400000  000100                        TTABLE  (TSSTAR,100)                    ; start up port
011164' 240001  011232'                       TTABLE  (TCHECK,1,TS33RS+3)             ; check if completed
011165' 340000  000000                        TTABLE  (TEXIT)                         ; exit if error yet

```

```
                                        ; Shifted 40 bits

011166' 140000  011224'                         TTABLE  (TCALL,TS33NN)          ; increment shift count
011167' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
011170' 240001  011233'                         TTABLE  (TCHECK,1,TS33RS+4)     ; check if completed
011171' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet
011172' 300000  011145'                         TTABLE  (TJRST,TS33L)           ; loop till done
011173' 000000  000000                          TTABLE  (TLAST)

                                        ; Initialize stuff

011174' 201 00 0 00 000000#             TS33IN: MOVEI   BUFF-1                  ; initialize buffer pointer
011175' 202 00 0 00 014034'                     MOVEM   T33BUF#                 ;  ...
011176' 263 17 0 00 000000                      RTN                             ; return

                                        ; Set up next data pattern

011177' 350 01 0 00 014034'             TS33NE: AOS     1,T33BUF                ; point to next data pattern
011200' 307 01 0 00 000000#                     CAIG    1,BUFF+5                ; done yet?
011201' 350 00 0 17 000000                      AOS     (P)                     ; no - set up proper return
011202' 200 00 0 01 000000                      MOVE    (1)                     ; get data pattern
011203' 201 02 0 00 011227'                     MOVEI   2,TS33RS                ; get initial address of EBUF data
011204' 242 00 0 00 777770              TS33NO: LSH     -8                      ; shift 4 bits
011205' 303 02 0 00 011227'                     CAILE   2,TS33RS                ; if past the first shift, insert
011206' 661 00 0 00 740000                      TLO     740000                  ;   the PLIN data into left 8 bits
011207' 202 00 0 02 000000                      MOVEM   (2)                     ; save it
011210' 350 00 0 00 000002                      AOS     2                       ; point to next address
011211' 307 02 0 00 011233'                     CAIG    2,TS33RS+4              ; done yet?
011212' 254 00 0 00 011204'                     JRST    TS33NO                  ; no - loop till done
011213' 402 00 0 00 014057'                     SETZM   TS33NM#                 ; initialize number of shifts

                                        ; Read the next word

011214' 201 01 0 00 000200                      MOVEI   1,200                   ; get start address
011215' 242 01 0 00 000001                      LSH     1,1                     ; position correctly
011216' 260 17 0 00 011124*                     GO      LDRAR                   ; load the register
011217' 201 01 0 00 000010                      MOVEI   1,MPRUN                 ; get MPRUN bit
011220' 260 17 0 00 011132*                     GO      LDCSR                   ; start up the port
011221' 400 01 0 00 000000                      SETZ    1,                      ; clear data
011222' 260 17 0 00 011220*                     GO      LDCSR                   ; stop the port
011223' 263 17 0 00 000000                      RTN                             ; return

                                        ; Increment shift count

011224' 201 00 0 00 000010              TS33NN: MOVEI   8                       ; get an 8
011225' 272 00 0 00 014057'                     ADDM    TS33NM                  ; add it
011226' 263 17 0 00 000000                      RTN                             ; return

011227'                                 TS33RS: BLOCK   5                       ; correct EBUF data
```

```
                                     ; Error messages

011234' 240000  011236'              MA33:   CALL!TXNOT!MA33PN              ; print data pattern ...
011235' 000000000000#                        LAST!CALL!TXALL!TTPNT          ; print test data

011236' 037 00 0 00 013336'          MA33PN: TMSGC    <Data >               ; get data and
011237' 200 01 0 00 014034'                  MOVE     1,T33BUF              ;   print it
011240' 200 00 0 01 000000                   MOVE     (1)
011241' 037 13 0 00 000000                   PNTHW
011242' 037 00 0 00 013502'                  TMSG     < shifted down from PLIN >
011243' 200 00 0 00 014057'                  MOVE     TS33NM                ; get shift count
011244' 037 15 0 00 000000                   PNTDEC
011245' 037 00 0 00 013343'                  TMSG     <. bits, result in EBUF>
011246' 263 17 0 00 000000                   RTN

                                     ; Microcode - Shift 8 bits

011247' 010001  010000               T33M:   MWORD    <ADDR=100,JMAP,J=101,D=1>                  ; 100
011250' 001000  000040
011251' 010100  000224                       MWORD    <CONT,D=1,SELF,MGC=224>                    ; 101
011252' 001000  003340
011253' 010200  000200                       MWORD    <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>         ; 102
011254' 732002  002340
011255' 010351  000000                       MWORD    <JMAP,J=5100,D=1>                          ; 103
011256' 001000  000040

                                     ;       Routine to read a word from CBUS into the formatter.

011257' 020002  010000                       MWORD    <ADDR=200,JMAP,J=201>                      ; 200
011260' 001000  000040
011261' 020102  030000                       MWORD    <CJP,J=203,D=1,CENA,CCCA>                  ; 201
011262' 001400  000060
011263' 020202  010000                       MWORD    <JMAP,J=201,D=1>                           ; 202
011264' 001000  000040
011265' 020300  000200                       MWORD    <CONT,D=1,SELF,MGC=200>                    ; 203
011266' 001000  003340
011267' 020400  000204                       MWORD    <CONT,D=1,SELF,MGC=204>                    ; 204
011270' 001000  003540
011271' 020500  000004                       MWORD    <CONT,D=1,SELC,MGC=4>                      ; 205
011272' 001000  004340
011273' 020600  000200                       MWORD    <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>         ; 206
011274' 732001  402340
011275' 020700  000000                       MWORD    <JZ>                                       ; 207
011276' 000000  000000

                                     ;       Initialization

011277' 100010  010020                       MWORD    <ADDR=1000,JMAP,J=1001,SELM,MGC=20>        ; 1000
011300' 000000  002040
011301' 100100  000200                       MWORD    <CONT,D=1,SELC,MGC=200>                    ; 1001
011302' 001000  004340
011303' 100200  310017                       MWORD    <LDCT,J=31,SDO,OR,D=2,SKCN,MGC=017>        ; 1002
011304' 732000  240300
011305' 100310  030000                       MWORD    <RPCT,J=1003,SOA,OR,D=5>                   ; 1003
```

```
 7670  011306' 435000  000220
 7671  011307' 100400  002040                    MWORD   <CONT,SOA,OR,D=1,OENA,SELM,MGC=40>               ; 1004
 7672  011310' 431000  002340
 7673  011311' 100500  000006                    MWORD   <JZ,SDO,D=1,SELP,MGC=6>                          ; 1005
 7674  011312' 701000  001000
 7675  011313' 000000  000000                    MWORD   <ADDR=0,JZ>                                      ; 0
 7676  011314' 000000  000000
 7677
 7678                                    ;       Initialization step
 7679
 7680  011315' 500050  010100                    MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>             ; 5000
 7681  011316' 000000  004040
 7682  011317' 500100  000020                    MWORD   <JZ,SELM,MGC=20>                                 ; 5001
 7683  011320' 000000  002000
 7684
 7685                                    ; Subroutine - Passed - Write 1 to bit 12 of CSR register
 7686
 7687  011321' 510000  000000                    MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>                 ; 5100
 7688  011322' 742001  000340
 7689  011323' 510100  000000                    MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                      ; 5101
 7690  011324' 302001  000740
 7691  011325' 510200  260000                    MWORD   <LDCT,J=26>                                      ; 5102
 7692  011326' 000000  000300
 7693  011327' 510351  030000                    MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>               ; 5103
 7694  011330' 102021  000220
 7695  011331' 510400  002004                    MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>            ; 5104
 7696  011332' 431020  005340
 7697  011333' 510551  070040                    MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>           ; 5105
 7698  011334' 001400  015060
 7699  011335' 510651  050000                    MWORD   <JMAP,J=5105,D=1>                                ; 5106
 7700  011336' 001000  000040
 7701  011337' 510700  000200                    MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>               ; 5107
 7702  011340' 431010  005340
 7703  011341' 511051  102004                    MWORD   <JMAP,J=5110,SOA,A=4,OR,D=1,OENA,SELE,MGC=4>     ; 5110
 7704  011342' 431040  005040
 7705  011343' 777777  777777                    -1
```

```

                                  ;#*********************************************************************
                                  ;* Test 34 - Fmtr Down Shift 8 (from PLIN) Test - SELRHTNIB=0
                                  ;
                                  ; Description:  Verify that the formatter can shift various  data
                                  ;               patterns down 8 bits at a  time  while  inserting
                                  ;               bits from the PLIN into the  formatter.  The data
                                  ;               loaded into the PLIN initially is 360.
                                  ;
                                  ; Procedure:    KL> Port Clear
                                  ;               KL> Set up data transfer for data patterns
                                  ;
                                  ;               UC> Read a data pattern over the CBUS
                                  ;               UC> Load PLIN with 360
                                  ;
                                  ;               UC> Shift down 8 bits
                                  ;               UC> Read formatter contents and write into EBUF
                                  ;               KL> Read EBUF and verify the data
                                  ;               KL> Start up the ucode to shift again
                                  ;
                                  ;               Repeat the above shifting then verifying EBUF
                                  ;                 contents 5 times, until the data becomes
                                  ;                 all zero.
                                  ;
                                  ;               Do for data patterns:   0's
                                  ;                                       1's
                                  ;                                       525252,,525252
                                  ;                                       000000,,007777
                                  ;                                       000000,,007070
                                  ;                                       000000,,000707
                                  ;
                                  ; Failure:      ---
                                  ;#*********************************************************************

                                  ; Test data

011344' 254 00 0 00 011354'       TSTC34: JRST    TG34                    ; go start test
011345' 100604  000034                    CBUS!NDMP!NDCB!ZCBUS!34         ; test mask
011346' 011532' 013521'                   T34M,,[ASCIZ ^Fmtr Down Shift 8 (from PLI[illegible] - SELRHTNIB=0^]
011347' 013263' 012716'                   [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
011350' 000000  011627'                   TSTC35                          ; failure test table
011351' 000000  012111'                   TSTC36                          ;   ...
011352' 000000  012372'                   TSTC37
011353' 777777  777777                    -1

                                  ; Start test

011354' 201 00 0 00 000000'       TG34:   MOVEI   Z9                      ; get address of module start
011355' 260 17 0 00 011072*               GO      TRACE                   ; handle trace output
011356' 201 01 0 00 011532'               MOVEI   1,T34M                  ; set up microcode address
011357' 260 17 0 00 011074*               GO      TLOAD                   ; load/verify it
011360' 263 17 0 00 000000                RTN                             ; failed - exit test
```

```

                                        ; Initialization

011361' 400 15 0 00 000000      TL34:   SETZ    ERFLG,                  ; clear error flag
011362' 260 17 0 00 011077*             GO      IPACLR                  ; clear port
011363' 402 00 0 00 011100*             SETZM   TSTSUB                  ; initialize subtest number
                                        MOVSI   [000000,,000000         ; get address of buffer
                                                 777777,,777777         ;   contents
                                                 525252,,525252
                                                 000000,,007777
                                                 000000,,007070
011364' 205 00 0 00 013330'                      000000,,000707]
011365' 541 00 0 00 011115*             HRRI    BUFF                    ; build the BLT pointer
011366' 251 00 0 00 000000#             BLT     BUFF+5                  ; insert data into buffer

                                        ; First ensure the data transfer is reset

011367' 201 01 0 00 005000              MOVEI   1,5000                  ; get start address
011370' 242 01 0 00 000001              LSH     1,1                     ; position correctly
011371' 260 17 0 00 011216*             GO      LDRAR                   ; load the register
011372' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
011373' 260 17 0 00 011222*             GO      LDCSR                   ; start up the port
011374' 400 01 0 00 000000              SETZ    1,                      ; clear data
011375' 260 17 0 00 011373*             GO      LDCSR                   ; stop the port

                                        ; Now generate CCW list

011376' 200 01 0 00 013210'             MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
011377' 260 17 0 00 011114*             GO      CHINIT                  ; initialize software
011400' 551 01 0 00 011365*             HRRZI   1,BUFF                  ; buffer address
011401' 201 02 0 00 000006              MOVEI   2,6                     ; word count
011402' 400 03 0 00 000000              SETZ    3,                      ; clear zero fill specifier
011403' 260 17 0 00 011120*             GO      GENCCW                  ; generate a CCW list
011404' 201 06 0 00 011427'             MOVEI   6,TS34                  ; get test execute table address

                                        ; Start up the data transfer

011405' 201 01 0 00 001000              MOVEI   1,1000                  ; get start address
011406' 242 01 0 00 000001              LSH     1,1                     ; position correctly
011407' 260 17 0 00 011371*             GO      LDRAR                   ; load the register
011410' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
011411' 260 17 0 00 011375*             GO      LDCSR                   ; start up the port
011412' 201 00 0 00 000001              MOVEI   1                       ; set up delay of 1 ms
011413' 260 17 0 00 011130*             GO      ODELAY                  ; wait a bit
011414' 400 01 0 00 000000              SETZ    1,                      ; clear data
011415' 260 17 0 00 011411*             GO      LDCSR                   ; stop the port

                                        ; Loop on test execute table entries

011416' 260 17 0 00 011133*     TA34:   GO      TEXEC                   ; execute table entry
011417' 254 00 0 00 011426'             JRST    TX34                    ; end of table
011420' 254 00 0 00 011416'             JRST    TA34                    ; keep looping after call
011421' 474 15 0 00 000000              SETO    ERFLG,                  ; error - EBUF data incorrect
```

```

                                        ; Handle error printouts and scope looping

011422' 027 00 0 00 011517'                     SCOPER  MA34                    ; print error message
011423' 254 00 0 00 011361'                     JRST    TL34                    ; loop on error
011424' 254 00 0 00 011426'                     JRST    TX34                    ; altmode exit
011425' 322 15 0 00 011416'                     JUMPE   ERFLG,TA34              ; do next sstep table entry

                                        ; End of test

011426' 263 17 0 00 000000              TX34:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

011427' 140000  011457'                 TS34:   TTABLE  (TCALL,TS34IN)          ; initialize stuff
011430' 200000  011462'                 TS34L:  TTABLE  (TCALLC,TS34NE)         ; set up next data pattern

                                        ; Shifted 8 bits

011431' 140000  011507'                 TS34A:  TTABLE  (TCALL,TS34NN)          ; increment shift count
011432' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
011433' 240001  011512'                         TTABLE  (TCHECK,1,TS34RS)       ; check if completed
011434' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 16 bits

011435' 140000  011507'                         TTABLE  (TCALL,TS34NN)          ; increment shift count
011436' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
011437' 240001  011513'                         TTABLE  (TCHECK,1,TS34RS+1)     ; check if completed
011440' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 24 bits

011441' 140000  011507'                         TTABLE  (TCALL,TS34NN)          ; increment shift count
011442' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
011443' 240001  011514'                         TTABLE  (TCHECK,1,TS34RS+2)     ; check if completed
011444' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 32 bits

011445' 140000  011507'                         TTABLE  (TCALL,TS34NN)          ; increment shift count
011446' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
011447' 240001  011515'                         TTABLE  (TCHECK,1,TS34RS+3)     ; check if completed
011450' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 40 bits

011451' 140000  011507'                         TTABLE  (TCALL,TS34NN)          ; increment shift count
011452' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
011453' 240001  011516'                         TTABLE  (TCHECK,1,TS34RS+4)     ; check if completed
011454' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet
011455' 300000  011430'                         TTABLE  (TJRST,TS34L)           ; loop till done
011456' 000000  000000                          TTABLE  (TLAST)

```

```

                                          ; Initialize stuff

011457' 201 00 0 00 000000#       TS34IN: MOVEI   BUFF-1                  ; initialize buffer pointer
011460' 202 00 0 00 014035'               MOVEM   T34BUF#                 ;  ...
011461' 263 17 0 00 000000                RTN                             ; return

                                          ; Set up next data pattern

011462' 350 01 0 00 014035'       TS34NE: AOS     1,T34BUF                ; point to next data pattern
011463' 307 01 0 00 000000#               CAIG    1,BUFF+5                ; done yet?
011464' 350 00 0 17 000000                AOS     (P)                     ; no - set up proper return
011465' 200 00 0 01 000000                MOVE    (1)                     ; get data pattern
011466' 201 02 0 00 011512'               MOVEI   2,TS34RS                ; get initial address of EBUF data
011467' 242 00 0 00 777770        TS34NO: LSH     -8                      ; shift 4 bits
011470' 303 02 0 00 011512'               CAILE   2,TS34RS                ; if past the first shift, insert
011471' 661 00 0 00 036000                TLO     36000                   ;   the PLIN data into left 8 bits
011472' 202 00 0 02 000000                MOVEM   (2)                     ; save it
011473' 350 00 0 00 000002                AOS     2                       ; point to next address
011474' 307 02 0 00 011516'               CAIG    2,TS34RS+4              ; done yet?
011475' 254 00 0 00 011467'               JRST    TS34NO                  ; no - loop till done
011476' 402 00 0 00 014060'               SETZM   TS34NM#                 ; initialize number of shifts

                                          ; Read the next word

011477' 201 01 0 00 000200                MOVEI   1,200                   ; get start address
011500' 242 01 0 00 000001                LSH     1,1                     ; position correctly
011501' 260 17 0 00 011407*               GO      LDRAR                   ; load the register
011502' 201 01 0 00 000010                MOVEI   1,MPRUN                 ; get MPRUN bit
011503' 260 17 0 00 011415*               GO      LDCSR                   ; start up the port
011504' 400 01 0 00 000000                SETZ    1,                      ; clear data
011505' 260 17 0 00 011503*               GO      LDCSR                   ; stop the port
011506' 263 17 0 00 000000                RTN                             ; return

                                          ; Increment shift count

011507' 201 00 0 00 000010        TS34NN: MOVEI   8                       ; get a 8
011510' 272 00 0 00 014060'               ADDM    TS34NM                  ; add it
011511' 263 17 0 00 000000                RTN                             ; return

011512'                           TS34RS: BLOCK   5                       ; correct EBUF data
```

```

                                      ; Error messages

011517' 240000  011521'               MA34:   CALL!TXNOT!MA34PN               ; print data pattern ...
011520' 000000000000#                         LAST!CALL!TXALL!TTPNT           ; print test data

011521' 037 00 0 00 013336'           MA34PN: TMSGC   <Data >                 ; get data and
011522' 200 01 0 00 014035'                   MOVE    1,T34BUF                ;   print it
011523' 200 00 0 01 000000                    MOVE    (1)
011524' 037 13 0 00 000000                    PNTHW
011525' 037 00 0 00 013502'                   TMSG    < shifted down from PLIN >
011526' 200 00 0 00 014060'                   MOVE    TS34NM                  ; get shift count
011527' 037 15 0 00 000000                    PNTDEC
011530' 037 00 0 00 013343'                   TMSG    <. bits, result in EBUF>
011531' 263 17 0 00 000000                    RTN

                                      ; Microcode - Shift 8 bits

011532' 010001  010000                T34M:   MWORD   <ADDR=100,JMAP,J=101,D=1>                       ; 100
011533' 001000  000040
011534' 010100  000224                        MWORD   <CONT,D=1,SELF,MGC=224>                         ; 101
011535' 001000  003340
011536' 010200  000200                        MWORD   <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>              ; 102
011537' 732002  002340
011540' 010351  000000                        MWORD   <JMAP,J=5100,D=1>                               ; 103
011541' 001000  000040

                                      ;       Routine to read a word from CBUS into the formatter.

011542' 020002  010000                        MWORD   <ADDR=200,JMAP,J=201>                           ; 200
011543' 001000  000040
011544' 020102  030000                        MWORD   <CJP,J=203,D=1,CENA,CCCA>                       ; 201
011545' 001400  000060
011546' 020202  010000                        MWORD   <JMAP,J=201,D=1>                                ; 202
011547' 001000  000040
011550' 020300  000200                        MWORD   <CONT,D=1,SELF,MGC=200>                         ; 203
011551' 001000  003340
011552' 020400  000204                        MWORD   <CONT,D=1,SELF,MGC=204>                         ; 204
011553' 001000  003340
011554' 020500  000004                        MWORD   <CONT,D=1,SELC,MGC=4>                           ; 205
011555' 001000  004340
011556' 020600  000200                        MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>              ; 206
011557' 732001  402340
011560' 020700  000000                        MWORD   <JZ>                                            ; 207
011561' 000000  000000

                                      ;       Initialization

011562' 100010  010020                        MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>             ; 1000
011563' 000000  002040
011564' 100100  000200                        MWORD   <CONT,D=1,SELC,MGC=200>                         ; 1001
011565' 001000  004340
011566' 100200  310360                        MWORD   <LDCT,J=31,SDO,OR,D=2,SKCN,MGC=360>             ; 1002
011567' 732000  240300
011570' 100310  030000                        MWORD   <RPCT,J=1003,SOA,OR,D=5>                        ; 1003
```

```
 7961  011571' 435000  000220
 7962  011572' 100400  002040                    MWORD   <CONT,SOA,OR,D=1,OENA,SELM,MGC=40>              ; 1004
 7963  011573' 431000  002340
 7964  011574' 100500  000006                    MWORD   <JZ,SDO,D=1,SELP,MGC=6>                         ; 1005
 7965  011575' 701000  001000
 7966  011576' 000000  000000                    MWORD   <ADDR=0,JZ>                                     ; 0
 7967  011577' 000000  000000
 7968
 7969                                    ;       Initialization step
 7970
 7971  011600' 500050  010100                    MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>            ; 5000
 7972  011601' 000000  004040
 7973  011602' 500100  000020                    MWORD   <JZ,SELM,MGC=20>                                ; 5001
 7974  011603' 000000  002000
 7975
 7976                                    ; Subroutine - Passed - Write 1 to bit 12 of CSR register
 7977
 7978  011604' 510000  000000                    MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>                ; 5100
 7979  011605' 742001  000340
 7980  011606' 510100  000000                    MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                     ; 5101
 7981  011607' 302001  000740
 7982  011610' 510200  260000                    MWORD   <LDCT,J=26>                                     ; 5102
 7983  011611' 000000  000300
 7984  011612' 510351  030000                    MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>              ; 5103
 7985  011613' 102021  000220
 7986  011614' 510400  002004                    MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>           ; 5104
 7987  011615' 431020  005340
 7988  011616' 510551  070040                    MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>          ; 5105
 7989  011617' 001400  015060
 7990  011620' 510651  050000                    MWORD   <JMAP,J=5105,D=1>                               ; 5106
 7991  011621' 001000  000040
 7992  011622' 510700  000200                    MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>              ; 5107
 7993  011623' 431010  005340
 7994  011624' 511051  102004                    MWORD   <JMAP,J=5110,SOA,A=4,OR,D=1,OENA,SELE,MGC=4>    ; 5110
 7995  011625' 431040  005040
 7996  011626' 777777  777777                    -1
```

```
                                           ;#*******************************************************************
                                           ;* Test 35 - Fmtr Down Shift 8 (from PLIN) Test - SELRHTNIB=1
                                           ;
                                           ; Description:  Verify that the formatter can shift various  data
                                           ;               patterns down 8 bits at a  time  while  inserting
                                           ;               bits from the PLIN into the  formatter.  The data
                                           ;               loaded into the PLIN initially is 017.
                                           ;
                                           ; Procedure:    KL> Port Clear
                                           ;               KL> Set up data transfer for data patterns
                                           ;
                                           ;               UC> Read a data pattern over the CBUS
                                           ;               UC> Load PLIN with 017
                                           ;
                                           ;               UC> Shift down 8 bits
                                           ;               UC> Read formatter contents and write into EBUF
                                           ;               KL> Read EBUF and verify the data
                                           ;               KL> Start up the ucode to shift again
                                           ;
                                           ;               Repeat the above shifting then verifying EBUF
                                           ;                 contents 5 times, until the data becomes
                                           ;                 all zero.
                                           ;
                                           ;               Do for data patterns:    0's
                                           ;                                        1's
                                           ;                                        525252,,525252
                                           ;                                        000000,,007777
                                           ;                                        000000,,007070
                                           ;                                        000000,,000707
                                           ;
                                           ; Failure:      ---
                                           ;#*******************************************************************

                                           ; Test data

011627' 254 00 0 00 011636'                TSTC35: JRST    TG35                    ; go start test
011630' 100604  000035                             CBUS!NDMP!NDCB!ZCBUS!35         ; test mask
011631' 012014' 013533'                            T35M,,[ASCIZ ^Fmtr Down Shift 8 (from PLIN) Test - SELRHTNIB=1^]
011632' 013263' 012716'                            [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
011633' 000000  012111'                            TSTC36                          ; failure test table
011634' 000000  012372'                            TSTC37                          ;   ...
011635' 777777  777777                             -1

                                           ; Start test

011636' 201 00 0 00 000000'                TG35:   MOVEI   Z9                      ; get address of module start
011637' 260 17 0 00 011355*                        GO      TRACE                   ; handle trace output
011640' 201 01 0 00 012014'                        MOVEI   1,T35M                  ; set up microcode address
011641' 260 17 0 00 011357*                        GO      TLOAD                   ; load/verify it
011642' 263 17 0 00 000000                         RTN                             ; failed - exit test
```

```

                                  ; Initialization

011643' 400 15 0 00 000000        TL35:   SETZ    ERFLG,                  ; clear error flag
011644' 260 17 0 00 011362*               GO      IPACLR                  ; clear port
011645' 402 00 0 00 011363*               SETZM   TSTSUB                  ; initialize subtest number
                                          MOVSI   [000000,,000000         ; get address of buffer
                                                   777777,,777777         ;   contents
                                                   525252,,525252
                                                   000000,,007777
                                                   000000,,007070
011646' 205 00 0 00 013330'                        000000,,000707]
011647' 541 00 0 00 011400*               HRRI    BUFF                    ; build the BLT pointer
011650' 251 00 0 00 000000#               BLT     BUFF+5                  ; insert data into buffer

                                  ; First ensure the data transfer is reset

011651' 201 01 0 00 005000                MOVEI   1,5000                  ; get start address
011652' 242 01 0 00 000001                LSH     1,1                     ; position correctly
011653' 260 17 0 00 011501*               GO      LDRAR                   ; load the register
011654' 201 01 0 00 000010                MOVEI   1,MPRUN                 ; get MPRUN bit
011655' 260 17 0 00 011505*               GO      LDCSR                   ; start up the port
011656' 400 01 0 00 000000                SETZ    1,                      ; clear data
011657' 260 17 0 00 011655*               GO      LDCSR                   ; stop the port

                                  ; Now generate CCW list

011660' 200 01 0 00 013210'               MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
011661' 260 17 0 00 011377*               GO      CHINIT                  ; initialize software
011662' 551 01 0 00 011647*               HRRZI   1,BUFF                  ; buffer address
011663' 201 02 0 00 000006                MOVEI   2,6                     ; word count
011664' 400 03 0 00 000000                SETZ    3,                      ; clear zero fill specifier
011665' 260 17 0 00 011403*               GO      GENCCW                  ; generate a CCW list
011666' 201 06 0 00 011711'               MOVEI   6,TS35                  ; get test execute table address

                                  ; Start up the data transfer

011667' 201 01 0 00 001000                MOVEI   1,1000                  ; get start address
011670' 242 01 0 00 000001                LSH     1,1                     ; position correctly
011671' 260 17 0 00 011653*               GO      LDRAR                   ; load the register
011672' 201 01 0 00 000010                MOVEI   1,MPRUN                 ; get MPRUN bit
011673' 260 17 0 00 011657*               GO      LDCSR                   ; start up the port
011674' 201 00 0 00 000001                MOVEI   1                       ; set up delay of 1 ms
011675' 260 17 0 00 011413*               GO      ODELAY                  ; wait a bit
011676' 400 01 0 00 000000                SETZ    1,                      ; clear data
011677' 260 17 0 00 011673*               GO      LDCSR                   ; stop the port
```

```

                                        ; Loop on test execute table entries

011700' 260 17 0 00 011416*     TA35:   GO      TEXEC                   ; execute table entry
011701' 254 00 0 00 011710'             JRST    TX35                    ; end of table
011702' 254 00 0 00 011700'             JRST    TA35                    ; keep looping after call
011703' 474 15 0 00 000000              SETO    ERFLG,                  ; error - EBUF data incorrect

                                        ; Handle error printouts and scope looping

011704' 027 00 0 00 012001'             SCOPER  MA35                    ; print error message
011705' 254 00 0 00 011643'             JRST    TL35                    ; loop on error
011706' 254 00 0 00 011710'             JRST    TX35                    ; altmode exit
011707' 322 15 0 00 011700'             JUMPE   ERFLG,TA35              ; do next sstep table entry

                                        ; End of test

011710' 263 17 0 00 000000      TX35:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

011711' 140000  011741'         TS35:   TTABLE  (TCALL,TS35IN)          ; initialize stuff
011712' 200000  011744'         TS35L:  TTABLE  (TCALLC,TS35NE)         ; set up next data pattern

                                        ; Shifted 8 bits

011713' 140000  011771'         TS35A:  TTABLE  (TCALL,TS35NN)          ; increment shift count
011714' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
011715' 240001  011774'                 TTABLE  (TCHECK,1,TS35RS)       ; check if completed
011716' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 16 bits

011717' 140000  011771'                 TTABLE  (TCALL,TS35NN)          ; increment shift count
011720' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
011721' 240001  011775'                 TTABLE  (TCHECK,1,TS35RS+1)     ; check if completed
011722' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 24 bits

011723' 140000  011771'                 TTABLE  (TCALL,TS35NN)          ; increment shift count
011724' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
011725' 240001  011776'                 TTABLE  (TCHECK,1,TS35RS+2)     ; check if completed
011726' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 32 bits

011727' 140000  011771'                 TTABLE  (TCALL,TS35NN)          ; increment shift count
011730' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
011731' 240001  011777'                 TTABLE  (TCHECK,1,TS35RS+3)     ; check if completed
011732' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

```

```
                                          ; Shifted 40 bits

011733' 140000  011771'                           TTABLE  (TCALL,TS35NN)          ; increment shift count
011734' 400000  000100                            TTABLE  (TSSTAR,100)            ; start up port
011735' 240001  012000'                           TTABLE  (TCHECK,1,TS35RS+4)     ; check if completed
011736' 340000  000000                            TTABLE  (TEXIT)                 ; exit if error yet
011737' 300000  011712'                           TTABLE  (TJRST,TS35L)           ; loop till done
011740' 000000  000000                            TTABLE  (TLAST)

                                          ; Initialize stuff

011741' 201 00 0 00 000000#               TS35IN: MOVEI   BUFF-1                  ; initialize buffer pointer
011742' 202 00 0 00 014036'                       MOVEM   T35BUF#                 ;  ...
011743' 263 17 0 00 000000                        RTN                             ; return

                                          ; Set up next data pattern

011744' 350 01 0 00 014036'               TS35NE: AOS     1,T35BUF                ; point to next data pattern
011745' 307 01 0 00 000000#                       CAIG    1,BUFF+5                ; done yet?
011746' 350 00 0 17 000000                        AOS     (P)                     ; no - set up proper return
011747' 200 00 0 01 000000                        MOVE    (1)                     ; get data pattern
011750' 201 02 0 00 011774'                       MOVEI   2,TS35RS                ; get initial address of EBUF data
011751' 242 00 0 00 777770                TS35NO: LSH     -8                      ; shift 4 bits
011752' 303 02 0 00 011774'                       CAILE   2,TS35RS                ; if past the first shift, insert
011753' 661 00 0 00 036000                        TLO     36000                   ;   the PLIN data into left 8 bits
011754' 202 00 0 02 000000                        MOVEM   (2)                     ; save it
011755' 350 00 0 00 000002                        AOS     2                       ; point to next address
011756' 307 02 0 00 012000'                       CAIG    2,TS35RS+4              ; done yet?
011757' 254 00 0 00 011751'                       JRST    TS35NO                  ; no - loop till done
011760' 402 00 0 00 014061'                       SETZM   TS35NM#                 ; initialize number of shifts

                                          ; Read the next word

011761' 201 01 0 00 000200                        MOVEI   1,200                   ; get start address
011762' 242 01 0 00 000001                        LSH     1,1                     ; position correctly
011763' 260 17 0 00 011671*                       GO      LDRAR                   ; load the register
011764' 201 01 0 00 000010                        MOVEI   1,MPRUN                 ; get MPRUN bit
011765' 260 17 0 00 011677*                       GO      LDCSR                   ; start up the port
011766' 400 01 0 00 000000                        SETZ    1,                      ; clear data
011767' 260 17 0 00 011765*                       GO      LDCSR                   ; stop the port
011770' 263 17 0 00 000000                        RTN                             ; return

                                          ; Increment shift count

011771' 201 00 0 00 000010                TS35NN: MOVEI   8                       ; get a 8
011772' 272 00 0 00 014061'                       ADDM    TS35NM                  ; add it
011773' 263 17 0 00 000000                        RTN                             ; return

011774'                                   TS35RS: BLOCK   5                       ; correct EBUF data
```

```


                                           ; Error messages

012001' 240000  012003'                  MA35:   CALL!TXNOT!MA35PN               ; print data pattern ...
012002' 000000000000#                            LAST!CALL!TXALL!TTPNT           ; print test data

012003' 037 00 0 00 013336'              MA35PN: TMSGC   <Data >                 ; get data and
012004' 200 01 0 00 014036'                      MOVE    1,T35BUF                ;   print it
012005' 200 00 0 01 000000                       MOVE    (1)
012006' 037 13 0 00 000000                       PNTHW
012007' 037 00 0 00 013502'                      TMSG    < shifted down from PLIN >
012010' 200 00 0 00 014061'                      MOVE    TS35NM                  ; get shift count
012011' 037 15 0 00 000000                       PNTDEC
012012' 037 00 0 00 013343'                      TMSG    <. bits, result in EBUF>
012013' 263 17 0 00 000000                       RTN

                                           ; Microcode - Shift 8 bits

012014' 010001  010000                   T35M:   MWORD   <ADDR=100,JMAP,J=101,D=1>                       ; 100
012015' 001000  000040
012016' 010100  000226                           MWORD   <CONT,D=1,SELF,MGC=226>                         ; 101
012017' 001000  003340
012020' 010200  000200                           MWORD   <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>              ; 102
012021' 732002  002340
012022' 010351  000000                           MWORD   <JMAP,J=5100,D=1>                               ; 103
012023' 001000  000040

                                           ;       Routine to read a word from CBUS into the formatter.

012024' 020002  010000                           MWORD   <ADDR=200,JMAP,J=201>                           ; 200
012025' 001000  000040
012026' 020102  030000                           MWORD   <CJP,J=203,D=1,CENA,CCCA>                       ; 201
012027' 001400  000060
012030' 020202  010000                           MWORD   <JMAP,J=201,D=1>                                ; 202
012031' 001000  000040
012032' 020300  000200                           MWORD   <CONT,D=1,SELF,MGC=200>                         ; 203
012033' 001000  003340
012034' 020400  000204                           MWORD   <CONT,D=1,SELF,MGC=204>                         ; 204
012035' 001000  003340
012036' 020500  000004                           MWORD   <CONT,D=1,SELC,MGC=4>                           ; 205
012037' 001000  004340
012040' 020600  000200                           MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>              ; 206
012041' 732001  402340
012042' 020700  000000                           MWORD   <JZ>                                            ; 207
012043' 000000  000000

                                           ;       Initialization

012044' 100010  010020                           MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>             ; 1000
012045' 000000  002040
012046' 100100  000200                           MWORD   <CONT,D=1,SELC,MGC=200>                         ; 1001
012047' 001000  004340
012050' 100200  310017                           MWORD   <LDCT,J=31,SDO,OR,D=2,SKCN,MGC=017>             ; 1002
012051' 732000  240300
012052' 100310  030000                           MWORD   <RPCT,J=1003,SOA,OR,D=5>                        ; 1003
```

```
012053' 435000  000220
012054' 100400  002040                  MWORD   <CONT,SOA,OR,D=1,OENA,SELM,MGC=40>                      ; 1004
012055' 431000  002340
012056' 100500  000006                  MWORD   <JZ,SDO,D=1,SELP,MGC=6>                                 ; 1005
012057' 701000  001000
012060' 000000  000000                  MWORD   <ADDR=0,JZ>                                             ; 0
012061' 000000  000000

                                ;       Initialization step

012062' 500050  010100                  MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>                    ; 5000
012063' 000000  004040
012064' 500100  000020                  MWORD   <JZ,SELM,MGC=20>                                        ; 5001
012065' 000000  002000

                                ; Subroutine - Passed - Write 1 to bit 12 of CSR register

012066' 510000  000000                  MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>                        ; 5100
012067' 742001  000340
012070' 510100  000000                  MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                             ; 5101
012071' 302001  000740
012072' 510200  260000                  MWORD   <LDCT,J=26>                                             ; 5102
012073' 000000  000300
012074' 510351  030000                  MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>                      ; 5103
012075' 102021  000220
012076' 510400  002004                  MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>                   ; 5104
012077' 431020  005340
012100' 510551  070040                  MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>                  ; 5105
012101' 001400  015060
012102' 510651  050000                  MWORD   <JMAP,J=5105,D=1>                                       ; 5106
012103' 001000  000040
012104' 510700  000200                  MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>                      ; 5107
012105' 431010  005340
012106' 511051  102004                  MWORD   <JMAP,J=5110,SOA,A=4,OR,D=1,OENA,SELE,MGC=4>            ; 5110
012107' 431040  005040
012110' 777777  777777                  -1
```

```

                                        ;#***********************************************************************
                                        ;* Test 36 - Fmtr Down Shift 8 (from PLIN) Test - SELRHTNIB=1
                                        ;
                                        ; Description:  Verify that the formatter can shift various  data
                                        ;               patterns down 4 bits at a  time  while  inserting
                                        ;               bits from the PLIN into the formatter.  The  data
                                        ;               loaded into the PLIN initially is 360.
                                        ;
                                        ; Procedure:    KL> Port Clear
                                        ;               KL> Set up data transfer for data patterns
                                        ;
                                        ;               UC> Read a data pattern over the CBUS
                                        ;               UC> Load PLIN with 360
                                        ;
                                        ;               UC> Shift down 4 bits
                                        ;               UC> Read formatter contents and write into EBUF
                                        ;               KL> Read EBUF and verify the data
                                        ;               KL> Start up the ucode to shift again
                                        ;
                                        ;               Repeat the above shifting then verifying EBUF
                                        ;                 contents 10 times, until the data becomes
                                        ;                 all zero.
                                        ;
                                        ;               Do for data patterns:   0's
                                        ;                                       1's
                                        ;                                       525252,,525252
                                        ;                                       000000,,007777
                                        ;                                       000000,,007070
                                        ;                                       000000,,000707
                                        ;
                                        ; Failure:      ---
                                        ;#***********************************************************************

                                        ; Test data

012111' 254 00 0 00 012117'             TSTC36: JRST    TG36                    ; go start test
012112' 100604  000036                          CBUS!NDMP!NDCB!ZCBUS!36         ; test mask
012113' 012275' 013507'                         T36M,,[ASCIZ ^Fmtr Down Shift 4 (from PLIN) Test - SELRHTNIB=1^]
012114' 013263' 012716'                         [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
012115' 000000  012372'                         TSTC37                          ; failure test table
012116' 777777  777777                          -1                              ;   ...

                                        ; Start test

012117' 201 00 0 00 000000'             TG36:   MOVEI   Z9                      ; get address of module start
012120' 260 17 0 00 011637*                     GO      TRACE                   ; handle trace output
012121' 201 01 0 00 012275'                     MOVEI   1,T36M                  ; set up microcode address
012122' 260 17 0 00 011641*                     GO      TLOAD                   ; load/verify it
012123' 263 17 0 00 000000                      RTN                             ; failed - exit test
```

```

                                        ; Initialization

012124' 400 15 0 00 000000              TL36:   SETZ    ERFLG,                  ; clear error flag
012125' 260 17 0 00 011644*                     GO      IPACLR                  ; clear port
012126' 402 00 0 00 011645*                     SETZM   TSTSUB                  ; initialize subtest number
                                                MOVSI   [000000,,000000         ; get address of buffer
                                                         777777,,777777         ;   contents
                                                         525252,,525252
                                                         000000,,007777
                                                         000000,,007070
012127' 205 00 0 00 013330'                              000000,,000707]
012130' 541 00 0 00 011662*                     HRRI    BUFF                    ; build the BLT pointer
012131' 251 00 0 00 000000#                     BLT     BUFF+5                  ; insert data into buffer

                                        ; First ensure the data transfer is reset

012132' 201 01 0 00 005000                      MOVEI   1,5000                  ; get start address
012133' 242 01 0 00 000001                      LSH     1,1                     ; position correctly
012134' 260 17 0 00 011763*                     GO      LDRAR                   ; load the register
012135' 201 01 0 00 000010                      MOVEI   1,MPRUN                 ; get MPRUN bit
012136' 260 17 0 00 011767*                     GO      LDCSR                   ; start up the port
012137' 400 01 0 00 000000                      SETZ    1,                      ; clear data
012140' 260 17 0 00 012136*                     GO      LDCSR                   ; stop the port

                                        ; Now generate CCW list

012141' 200 01 0 00 013210'                     MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
012142' 260 17 0 00 011661*                     GO      CHINIT                  ; initialize software
012143' 551 01 0 00 012130*                     HRRZI   1,BUFF                  ; buffer address
012144' 201 02 0 00 000006                      MOVEI   2,6                     ; word count
012145' 400 03 0 00 000000                      SETZ    3,                      ; clear zero fill specifier
012146' 260 17 0 00 011665*                     GO      GENCCW                  ; generate a CCW list
012147' 201 06 0 00 012172'                     MOVEI   6,TS36                  ; get test execute table address

                                        ; Start up the data transfer

012150' 201 01 0 00 001000                      MOVEI   1,1000                  ; get start address
012151' 242 01 0 00 000001                      LSH     1,1                     ; position correctly
012152' 260 17 0 00 012134*                     GO      LDRAR                   ; load the register
012153' 201 01 0 00 000010                      MOVEI   1,MPRUN                 ; get MPRUN bit
012154' 260 17 0 00 012140*                     GO      LDCSR                   ; start up the port
012155' 201 00 0 00 000001                      MOVEI   1                       ; set up delay of 1 ms
012156' 260 17 0 00 011675*                     GO      ODELAY                  ; wait a bit
012157' 400 01 0 00 000000                      SETZ    1,                      ; clear data
012160' 260 17 0 00 012154*                     GO      LDCSR                   ; stop the port

                                        ; Loop on test execute table entries

012161' 260 17 0 00 011700*             TA36:   GO      TEXEC                   ; execute table entry
012162' 254 00 0 00 012171'                     JRST    TX36                    ; end of table
012163' 254 00 0 00 012161'                     JRST    TA36                    ; keep looping after call
012164' 474 15 0 00 000000                      SETO    ERFLG,                  ; error - EBUF data incorrect
```

```

                                        ; Handle error printouts and scope looping

012165' 027 00 0 00 012262'                     SCOPER  MA36                    ; print error message
012166' 254 00 0 00 012124'                     JRST    TL36                    ; loop on error
012167' 254 00 0 00 012171'                     JRST    TX36                    ; altmode exit
012170' 322 15 0 00 012161'                     JUMPE   ERFLG,TA36              ; do next sstep table entry

                                        ; End of test

012171' 263 17 0 00 000000              TX36:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

012172' 140000  012222'                 TS36:   TTABLE  (TCALL,TS36IN)          ; initialize stuff
012173' 200000  012225'                 TS36L:  TTABLE  (TCALLC,TS36NE)         ; set up next data pattern

                                        ; Shifted 8 bits

012174' 140000  012252'                 TS36A:  TTABLE  (TCALL,TS36NN)          ; increment shift count
012175' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
012176' 240001  012255'                         TTABLE  (TCHECK,1,TS36RS)       ; check if completed
012177' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 16 bits

012200' 140000  012252'                         TTABLE  (TCALL,TS36NN)          ; increment shift count
012201' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
012202' 240001  012256'                         TTABLE  (TCHECK,1,TS36RS+1)     ; check if completed
012203' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 24 bits

012204' 140000  012252'                         TTABLE  (TCALL,TS36NN)          ; increment shift count
012205' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
012206' 240001  012257'                         TTABLE  (TCHECK,1,TS36RS+2)     ; check if completed
012207' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

                                        ; Shifted 32 bits

012210' 140000  012252'                         TTABLE  (TCALL,TS36NN)          ; increment shift count
012211' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
012212' 240001  012260'                         TTABLE  (TCHECK,1,TS36RS+3)     ; check if completed
012213' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet

```

```

                                        ; Shifted 40 bits

012214' 140000  012252'                         TTABLE  (TCALL,TS36NN)          ; increment shift count
012215' 400000  000100                          TTABLE  (TSSTAR,100)            ; start up port
012216' 240001  012261'                         TTABLE  (TCHECK,1,TS36RS+4)     ; check if completed
012217' 340000  000000                          TTABLE  (TEXIT)                 ; exit if error yet
012220' 300000  012173'                         TTABLE  (TJRST,TS36L)           ; loop till done
012221' 000000  000000                          TTABLE  (TLAST)

                                        ; Initialize stuff

012222' 201 00 0 00 000000#             TS36IN: MOVEI   BUFF-1                  ; initialize buffer pointer
012223' 202 00 0 00 014037'                     MOVEM   T36BUF#                 ;  ...
012224' 263 17 0 00 000000                      RTN                             ; return

                                        ; Set up next data pattern

012225' 350 01 0 00 014037'             TS36NE: AOS     1,T36BUF                ; point to next data pattern
012226' 307 01 0 00 000000#                     CAIG    1,BUFF+5                ; done yet?
012227' 350 00 0 17 000000                      AOS     (P)                     ; no - set up proper return
012230' 200 00 0 01 000000                      MOVE    (1)                     ; get data pattern
012231' 201 02 0 00 012255'                     MOVEI   2,TS36RS                ; get initial address of EBUF data
012232' 242 00 0 00 777770              TS36N0: LSH     -8                      ; shift 4 bits
012233' 303 02 0 00 012255'                     CAILE   2,TS36RS                ; if past the first shift, insert
012234' 661 00 0 00 740000                      TLO     740000                  ;   the PLIN data into left 8 bits
012235' 202 00 0 02 000000                      MOVEM   (2)                     ; save it
012236' 350 00 0 00 000002                      AOS     2                       ; point to next address
012237' 307 02 0 00 012261'                     CAIG    2,TS36RS+4              ; done yet?
012240' 254 00 0 00 012232'                     JRST    TS36N0                  ; no - loop till done
012241' 402 00 0 00 014062'                     SETZM   TS36NM#                 ; initialize number of shifts

                                        ; Read the next word

012242' 201 01 0 00 000200                      MOVEI   1,200                   ; get start address
012243' 242 01 0 00 000001                      LSH     1,1                     ; position correctly
012244' 260 17 0 00 012152*                     GO      LDRAR                   ; load the register
012245' 201 01 0 00 000010                      MOVEI   1,MPRUN                 ; get MPRUN bit
012246' 260 17 0 00 012160*                     GO      LDCSR                   ; start up the port
012247' 400 01 0 00 000000                      SETZ    1,                      ; clear data
012250' 260 17 0 00 012246*                     GO      LDCSR                   ; stop the port
012251' 263 17 0 00 000000                      RTN                             ; return

                                        ; Increment shift count

012252' 201 00 0 00 000010              TS36NN: MOVEI   8                       ; get a 8
012253' 272 00 0 00 014062'                     ADDM    TS36NM                  ; add it
012254' 263 17 0 00 000000                      RTN                             ; return

012255'                                 TS36RS: BLOCK   5                       ; correct EBUF data
```

```

                                  ; Error messages

012262' 240000  012264'           MA36:   CALL!TXNOT!MA36PN             ; print data pattern ...
012263' 000000000000#                     LAST!CALL!TXALL!TTPNT         ; print test data

012264' 037 00 0 00 013336'       MA36PN: TMSGC   <Data >               ; get data and
012265' 200 01 0 00 014037'               MOVE    1,T36BUF              ;   print it
012266' 200 00 0 01 000000                MOVE    (1)
012267' 037 13 0 00 000000                PNTHW
012270' 037 00 0 00 013502'               TMSG    < shifted down from PLIN >
012271' 200 00 0 00 014062'               MOVE    TS36NM                ; get shift count
012272' 037 15 0 00 000000                PNTDEC
012273' 037 00 0 00 013343'               TMSG    <. bits, result in EBUF>
012274' 263 17 0 00 000000                RTN

                                  ; Microcode - Shift 8 bits

012275' 010001  010000            T36M:   MWORD   <ADDR=100,JMAP,J=101,D=1>                   ; 100
012276' 001000  000040
012277' 010100  000226                    MWORD   <CONT,D=1,SELF,MGC=226>                     ; 101
012300' 001000  003340
012301' 010200  000200                    MWORD   <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>          ; 102
012302' 732002  002340
012303' 010351  000000                    MWORD   <JMAP,J=5100,D=1>                           ; 103
012304' 001000  000040

                                  ;       Routine to read a word from CBUS into the formatter.

012305' 020002  010000                    MWORD   <ADDR=200,JMAP,J=201>                       ; 200
012306' 001000  000040
012307' 020102  030000                    MWORD   <CJP,J=203,D=1,CENA,CCCA>                   ; 201
012310' 001400  000060
012311' 020202  010000                    MWORD   <JMAP,J=201,D=1>                            ; 202
012312' 001000  000040
012313' 020300  000200                    MWORD   <CONT,D=1,SELF,MGC=200>                     ; 203
012314' 001000  003340
012315' 020400  000204                    MWORD   <CONT,D=1,SELF,MGC=204>                     ; 204
012316' 001000  003340
012317' 020500  000004                    MWORD   <CONT,D=1,SELC,MGC=4>                       ; 205
012320' 001000  004340
012321' 020600  000200                    MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>          ; 206
012322' 732001  402340
012323' 020700  000000                    MWORD   <JZ>                                        ; 207
012324' 000000  000000

                                  ;       Initialization

012325' 100010  010020                    MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>         ; 1000
012326' 000000  002040
012327' 100100  000200                    MWORD   <CONT,D=1,SELC,MGC=200>                     ; 1001
012330' 001000  004340
012331' 100200  310360                    MWORD   <LDCT,J=31,SDO,OR,D=2,SKCN,MGC=360>         ; 1002
012332' 732000  240300
012333' 100310  030000                    MWORD   <RPCT,J=1003,SOA,OR,D=5>                    ; 1003
```

```
 8540  012334' 435000  000220
 8541  012335' 100400  002040                    MWORD   <CONT,SOA,OR,D=1,OENA,SELM,MGC=40>                 ; 1004
 8542  012336' 431000  002340
 8543  012337' 100500  000006                    MWORD   <JZ,SDO,D=1,SELP,MGC=6>                            ; 1005
 8544  012340' 701000  001000
 8545  012341' 000000  000000                    MWORD   <ADDR=0,JZ>                                        ; 0
 8546  012342' 000000  000000
 8547
 8548                                    ;       Initialization step
 8549
 8550  012343' 500050  010100                    MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>               ; 5000
 8551  012344' 000000  004040
 8552  012345' 500100  000020                    MWORD   <JZ,SELM,MGC=20>                                   ; 5001
 8553  012346' 000000  002000
 8554
 8555                                    ; Subroutine - Passed - Write 1 to bit 12 of CSR register
 8556
 8557  012347' 510000  000000                    MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>                   ; 5100
 8558  012350' 742001  000340
 8559  012351' 510100  000000                    MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                        ; 5101
 8560  012352' 302001  000740
 8561  012353' 510200  260000                    MWORD   <LDCT,J=26>                                        ; 5102
 8562  012354' 000000  000300
 8563  012355' 510351  030000                    MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>                 ; 5103
 8564  012356' 102021  000220
 8565  012357' 510400  002004                    MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>              ; 5104
 8566  012360' 431020  005340
 8567  012361' 510551  070040                    MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>             ; 5105
 8568  012362' 001400  015060
 8569  012363' 510651  050000                    MWORD   <JMAP,J=5105,D=1>                                  ; 5106
 8570  012364' 001000  000040
 8571  012365' 510700  000200                    MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>                 ; 5107
 8572  012366' 431010  005340
 8573  012367' 511051  102004                    MWORD   <JMAP,J=5110,SOA,A=4,OR,D=1,OENA,SELE,MGC=4>       ; 5110
 8574  012370' 431040  005040
 8575  012371' 777777  777777                    -1
```

```
                                        ;#*********************************************************************
                                        ;* Test 37 - Fmtr Ring Buffer Test
                                        ;
                                        ; Description:  Verify that the formatter is capable of  shifting
                                        ;               down in a circular manner, retaining the data  in
                                        ;               the formatter.
                                        ;
                                        ; Procedure:    KL> Port Clear
                                        ;               KL> Set up data transfer for data patterns
                                        ;
                                        ;               UC> Read a data pattern over the CBUS
                                        ;               UC> Shift formatter up/down 8 bits to clear it
                                        ;               UC> Load data into formatter
                                        ;               UC> Shift down 4 bits (11 times)
                                        ;               UC> Read formatter contents and verify same
                                        ;                   as it was originally
                                        ;               UC> Repeat the above 2 steps 512 times
                                        ;
                                        ;               Repeat the above 6 steps 6 times, for each data
                                        ;                 pattern.
                                        ;
                                        ;               Do for data patterns:   0's
                                        ;                                       1's
                                        ;                                       525252,,525252
                                        ;                                       000000,,007777
                                        ;                                       000000,,007070
                                        ;                                       000000,,000707
                                        ;
                                        ; Failure:      ---
                                        ;#*********************************************************************

                                        ; Test data

 012372' 254 00 0 00 012377'            TSTC37: JRST    TG37                    ; go start test
 012373' 100604  000037                         CBUS!NDMP!NDCB!ZCBUS!37         ; test mask
 012374' 012536' 013545'                        T37M,,[ASCIZ ^Fmtr Ring Buffer Test^]
 012375' 013263' 012716'                        [EXP C20,C12,MLAST!C8],,[EXP C4,MLAST!C2]
 012376' 777777  777777                         -1                              ; failure test table

                                        ; Start test

 012377' 201 00 0 00 000000'            TG37:   MOVEI   Z9                      ; get address of module start
 012400' 260 17 0 00 012120*                    GO      TRACE                   ; handle trace output
 012401' 201 01 0 00 012536'                    MOVEI   1,T37M                  ; set up microcode address
 012402' 260 17 0 00 012122*                    GO      TLOAD                   ; load/verify it
 012403' 263 17 0 00 000000                     RTN                             ; failed - exit test
```

```

                                ; Initialization

012404' 400 15 0 00 000000      TL37:   SETZ    ERFLG,                  ; clear error flag
012405' 260 17 0 00 012125*             GO      IPACLR                  ; clear port
012406' 402 00 0 00 012126*             SETZM   TSTSUB                  ; initialize subtest number
                                        MOVSI   [000000,,000000         ; get address of buffer
                                                 777777,,777777         ;   contents
                                                 525252,,525252
                                                 000000,,007777
                                                 000000,,007070
012407' 205 00 0 00 013330'                      000000,,000707]
012410' 541 00 0 00 012143*             HRRI    BUFF                    ; build the BLT pointer
012411' 251 00 0 00 000000#             BLT     BUFF+5                  ; insert data into buffer

                                ; First ensure the data transfer is reset

012412' 201 01 0 00 005000              MOVEI   1,5000                  ; get start address
012413' 242 01 0 00 000001              LSH     1,1                     ; position correctly
012414' 260 17 0 00 012244*             GO      LDRAR                   ; load the register
012415' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
012416' 260 17 0 00 012250*             GO      LDCSR                   ; start up the port
012417' 201 00 0 00 000001              MOVEI   1                       ; wait a little bit
012420' 260 17 0 00 012156*             GO      ODELAY                  ;   (1 millisecond)
012421' 400 01 0 00 000000              SETZ    1,                      ; clear data
012422' 260 17 0 00 012416*             GO      LDCSR                   ; stop the port

                                ; Now generate CCW list

012423' 200 01 0 00 013210'             MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
012424' 260 17 0 00 012142*             GO      CHINIT                  ; initialize software
012425' 551 01 0 00 012410*             HRRZI   1,BUFF                  ; buffer address
012426' 201 02 0 00 000006              MOVEI   2,6                     ; word count
012427' 400 03 0 00 000000              SETZ    3,                      ; clear zero fill specifier
012430' 260 17 0 00 012146*             GO      GENCCW                  ; generate a CCW list
012431' 201 06 0 00 012477'             MOVEI   6,TS37                  ; get test execute table address

                                ; Start up the data transfer

012432' 201 01 0 00 001000              MOVEI   1,1000                  ; get start address
012433' 242 01 0 00 000001              LSH     1,1                     ; position correctly
012434' 260 17 0 00 012414*             GO      LDRAR                   ; load the register
012435' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
012436' 260 17 0 00 012422*             GO      LDCSR                   ; start up the port
012437' 201 00 0 00 000001              MOVEI   1                       ; wait a little bit
012440' 260 17 0 00 012420*             GO      ODELAY                  ;   (1 millisecond)
012441' 400 01 0 00 000000              SETZ    1,                      ; clear data
012442' 260 17 0 00 012436*             GO      LDCSR                   ; stop the port
```

```

                                        ; Loop on test execute table entries

012443' 260 17 0 00 012161*     TA37:   GO      TEXEC                   ; execute table entry
012444' 254 00 0 00 012476'             JRST    TX37                    ; end of table
012445' 254 00 0 00 012443'             JRST    TA37                    ; keep looping after call
012446' 474 15 0 00 000000              SETO    ERFLG,                  ; error - EBUF data incorrect

                                        ; Get correct/actual data (R3/R4)

012447' 260 17 0 00 012405*             GO      IPACLR                  ; do a 'port clear'
012450' 201 01 0 00 002000              MOVEI   1,2000                  ; set start address to
012451' 242 01 0 00 000001              LSH     1,1                     ;   2000
012452' 260 17 0 00 012434*             GO      LDRAR
012453' 201 01 0 00 000010              MOVEI   1,MPRUN
012454' 260 17 0 00 012442*             GO      LDCSR                   ; start the port
012455' 201 01 0 00 200000              MOVEI   1,TSTEBF
012456' 260 17 0 00 012454*             GO      LDCSR                   ; stop the port
012457' 260 17 0 00 007021*             GO      RDEBUF                  ; read EBUF data
012460' 202 01 0 00 000000*             MOVEM   1,TEBUFC                ; save it
012461' 201 01 0 00 002002              MOVEI   1,2002                  ; set start address to
012462' 242 01 0 00 000001              LSH     1,1                     ;   2002
012463' 260 17 0 00 012452*             GO      LDRAR
012464' 201 01 0 00 000010              MOVEI   1,MPRUN
012465' 260 17 0 00 012456*             GO      LDCSR                   ; start the port
012466' 201 01 0 00 200000              MOVEI   1,TSTEBF
012467' 260 17 0 00 012465*             GO      LDCSR                   ; stop the port
012470' 260 17 0 00 012457*             GO      RDEBUF                  ; read EBUF data
012471' 202 01 0 00 000000*             MOVEM   1,TEBUFA                ; save it

                                        ; Handle error printouts and scope looping

012472' 027 00 0 00 012525'             SCOPER  MA37                    ; print error message
012473' 254 00 0 00 012404'             JRST    TL37                    ; loop on error
012474' 254 00 0 00 012476'             JRST    TX37                    ; altmode exit
012475' 322 15 0 00 012443'             JUMPE   ERFLG,TA37              ; do next sstep table entry

                                        ; End of test

012476' 263 17 0 00 000000      TX37:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

012477' 140000  012506'         TS37:   TTABLE  (TCALL,TS37IN)          ; init data pattern
012500' 200000  012510'         TS37A:  TTABLE  (TCALLC,TS37CH)         ; set up next data pattern
012501' 400000  000100                  TTABLE  (TSSTAR,100)            ; start up port
012502' 240005  000000                  TTABLE  (TCHECK,5,0)            ; check if completed
012503' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet
012504' 300000  012500'                 TTABLE  (TJRST,TS37A)           ; loop till done
012505' 000000  000000                  TTABLE  (TLAST)

```

```

012506' 476 00 0 00 014063'     TS37IN: SETOM   TS37DA#                 ; init word number
012507' 263 17 0 00 000000              RTN                             ; return

012510' 350 01 0 00 014063'     TS37CH: AOS     1,TS37DA                ; point to next word number
012511' 305 01 0 00 000006              CAIGE   1,6                     ; done yet?
012512' 350 00 0 17 000000              AOS     (P)                     ; no - set up proper return

                                ; Read the next word

012513' 201 01 0 00 000200              MOVEI   1,200                   ; get start address
012514' 242 01 0 00 000001              LSH     1,1                     ; position correctly
012515' 260 17 0 00 012463*             GO      LDRAR                   ; load the register
012516' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
012517' 260 17 0 00 012467*             GO      LDCSR                   ; start up the port
012520' 201 00 0 00 000001              MOVEI   1                       ; wait a little bit
012521' 260 17 0 00 012440*             GO      ODELAY                  ;   (1 millisecond)
012522' 400 01 0 00 000000              SETZ    1,                      ; clear data
012523' 260 17 0 00 012517*             GO      LDCSR                   ; stop the port
012524' 263 17 0 00 000000              RTN                             ; return

                                ; Error messages

012525' 240000  012527'         MA37:   CALL!TXNOT!MA37PN               ; print data pattern ...
012526' 000000000000#                   LAST!CALL!TXALL!TTPNT           ; print test data

012527' 037 00 0 00 013336'     MA37PN: TMSGC   <Data >                 ; get data and
012530' 200 01 0 00 014063'             MOVE    1,TS37DA                ;   print it
012531' 200 00 0 01 012425*             MOVE    BUFF(1)
012532' 037 13 0 00 000000              PNTHW
012533' 037 00 0 00 013552'             TMSG    < changed while being shifted down continuously.>
012534' 037 00 0 00 013564'             TMSGC   <Result in EBUF.>
012535' 263 17 0 00 000000              RTN

                                ; Microcode

012536' 010001  010000          T37M:   MWORD   <ADDR=100,JMAP,J=101,SOB,AND,B=5,D=2>           ; 100
012537' 342002  400040
012540' 010110  000000                  MWORD   <LDCT,J=1000>                                   ; 101
012541' 000000  000300
012542' 010200  000204                  MWORD   <CONT,SOB,PLUS,B=5,D=2,CRY,SELF,MGC=204>        ; 102
012543' 302002  403740
012544' 010300  000104                  MWORD   <CONT,SELF,MGC=104>                             ; 103
012545' 000000  003340
012546' 010400  000204                  MWORD   <CONT,SELF,MGC=204>                             ; 104
012547' 000000  003340
012550' 010500  000104                  MWORD   <CONT,SELF,MGC=104>                             ; 105
012551' 000000  003340
012552' 010600  000204                  MWORD   <CONT,SELF,MGC=204>                             ; 106
012553' 000000  003340
012554' 010700  000104                  MWORD   <CONT,SELF,MGC=104>                             ; 107
012555' 000000  003340
012556' 011000  000204                  MWORD   <CONT,SELF,MGC=204>                             ; 110
012557' 000000  003340
012560' 011100  000104                  MWORD   <CONT,SELF,MGC=104>                             ; 111
```

```
012561' 000000  003340
012562' 011200  000200                  MWORD   <CONT,SDO,OR,B=4,D=2,SELM,MGC=200>                      ; 112
012563' 732002  002340
012564' 011301  150000                  MWORD   <CJP,J=115,SAB,A=3,B=4,D=1,SMIN,CRY,CENA,CCFZ>          ; 113
012565' 111432  020460
012566' 011452  000000                  MWORD   <JMAP,J=5200,D=1>                                       ; 114
012567' 001000  000040
012570' 011501  020000                  MWORD   <RPCT,J=102>                                            ; 115
012571' 000000  000220
012572' 011651  000000                  MWORD   <JMAP,J=5100,D=1>                                       ; 116
012573' 001000  000040

                                ;       Routine to read a word from CBUS into the formatter.

012574' 020002  010000                  MWORD   <ADDR=200,JMAP,J=201>                                   ; 200
012575' 000000  000040
012576' 020102  030000                  MWORD   <CJP,J=203,D=1,CENA,CCCA>                               ; 201
012577' 001400  000060
012600' 020202  010000                  MWORD   <JMAP,J=201,D=1>                                        ; 202
012601' 001000  000040
012602' 020300  000200                  MWORD   <CONT,D=1,SELF,MGC=200>                                 ; 203
012603' 001000  003340
012604' 020400  000204                  MWORD   <CONT,D=1,SELF,MGC=204>                                 ; 204
012605' 001000  003340
012606' 020500  000004                  MWORD   <CONT,D=1,SELC,MGC=4>                                   ; 205
012607' 001000  004340
012610' 020600  000200                  MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>                      ; 206
012611' 732001  402340
012612' 020700  000000                  MWORD   <JZ,SOA,OR,A=3,B=6,D=2>                                 ; 207
012613' 432033  000000

                                ;       Initialization

012614' 100010  010020                  MWORD   <ADDR=1000,JMAP,J=1001,SELM,MGC=20>                     ; 1000
012615' 000000  002040
012616' 100100  000200                  MWORD   <JZ,D=1,SELC,MGC=200>                                   ; 1001
012617' 001000  004000
012620' 000000  000000                  MWORD   <ADDR=0,JZ>                                             ; 0
012621' 000000  000000

                                ;       Read correct/actual data

012622' 200020  012004                  MWORD   <ADDR=2000,JMAP,J=2001,SOA,A=3,OR,D=2,OENA,SELE,MGC=4>
012623' 432030  005040
012624' 200120  010000                  MWORD   <JMAP,J=2001>                                           ; 2001
012625' 000000  000040
012626' 200220  032004                  MWORD   <JMAP,J=2003,SOA,A=3,OR,D=2,OENA,SELE,MGC=4>            ; 2002
012627' 432030  005040
012630' 200320  030000                  MWORD   <JMAP,J=2003>                                           ; 2003
012631' 000000  000040

                                ;       Initialization step

012632' 500050  010100                  MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>                    ; 5000
012633' 000000  004040
```

```
 8832  012634' 500100  000020               MWORD   <JZ,SELM,MGC=20>                                ; 5001
 8833  012635' 000000  002000
 8834
 8835                                 ; Subroutine - Passed - Write 1 to bit 12 of CSR register
 8836
 8837  012636' 510000  000000               MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>                ; 5100
 8838  012637' 742001  000340
 8839  012640' 510100  000000               MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                     ; 5101
 8840  012641' 302001  000740
 8841  012642' 510200  260000               MWORD   <LDCT,J=26>                                     ; 5102
 8842  012643' 000000  000300
 8843  012644' 510351  030000               MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>              ; 5103
 8844  012645' 102021  000220
 8845  012646' 510400  002004               MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>           ; 5104
 8846  012647' 431020  005340
 8847  012650' 510551  070040               MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>          ; 5105
 8848  012651' 001400  015060
 8849  012652' 510651  050000               MWORD   <JMAP,J=5105,D=1>                               ; 5106
 8850  012653' 001000  000040
 8851  012654' 510700  000200               MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>              ; 5107
 8852  012655' 431010  005340
 8853  012656' 511051  102004               MWORD   <JMAP,J=5110,SOA,OR,A=4,D=1,OENA,SELE,MGC=4>    ; 5110
 8854  012657' 431040  005040
 8855
 8856                                 ; Subroutine - Failed - Write 1 to bit 13 of CSR register
 8857
 8858  012660' 520000  000000               MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>                ; 5200
 8859  012661' 742001  000340
 8860  012662' 520100  000000               MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                     ; 5201
 8861  012663' 302001  000740
 8862  012664' 520200  250000               MWORD   <LDCT,J=25>                                     ; 5202
 8863  012665' 000000  000300
 8864  012666' 520352  030000               MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>              ; 5203
 8865  012667' 102021  000220
 8866  012670' 520400  002004               MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>           ; 5204
 8867  012671' 431020  005340
 8868  012672' 520552  070040               MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>          ; 5205
 8869  012673' 001400  015060
 8870  012674' 520652  050000               MWORD   <JMAP,J=5205,D=1>                               ; 5206
 8871  012675' 001000  000040
 8872  012676' 520700  000200               MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>              ; 5207
 8873  012677' 431010  005340
 8874  012700' 521052  102004               MWORD   <JMAP,J=5210,SOA,OR,A=4,D=1,OENA,SELE,MGC=4>    ; 5210
 8875  012701' 431040  005040
 8876  012702' 777777  777777               -1
 8877
 8878
 8879                                 ;#***********************************************************************
 8880                                 ; End of CBUS Tests (Part 1)
 8881                                 ;#***********************************************************************
 8882
 8883                                         XLIST
 8884
NO ERRORS DETECTED
```

```
PROGRAM BREAK IS 014065
CPU TIME USED 07:37.654

207P CORE USED
```

```
BBPNT          000000  ext    IPACLR         012447' ext    MA37           012525'        T16M           004542'
BEXEC          003056' ext    IPASRT         007000' ext    MA37PN         012527'        T17BUF         014021'
BUFCOM         002642' ext    LAST    010000 000000  spd    MA4            001216'        T17M           005053'
BUFF           012531' ext    LDCSR          012523' ext    MA5            001370'        T1M            000152'
BUFGEN         000000  ext    LDRAR          012515' ext    MA5CHO         001406'        T20BUF         014022'
C1             000054  spd    MA1            000150'        MA5CHE         001373'        T20M           005365'
C12            000067  spd    MA10           002632'        MA6            001742'        T21BUF         014023'
C13            000070  spd    MA10CH         002635'        MA6CHO         001760'        T21M           005645'
C14            000071  spd    MA11           003072'        MA6CH1         002006'        T22BUF         014024'
C15            000072  spd    MA12           003155'        MA6CHE         001745'        T22M           006124'
C16            000073  spd    MA12P1         003157'        MA7            002321'        T23BUF         014025'
C17            000074  spd    MA12PN         003171'        MA7CHE         002324'        T23M           006402'
C18            000075  spd    MA13           003414'        MB12           003167'        T24BUF         014026'
C19            000076  spd    MA13PN         003416'        MB25           007041'        T24M           006663'
C2             000055  spd    MA14           003713'        MBCN           000016         T25M           007047'
C20            000077  spd    MA14PN         003715'        MBUS    010000 000000  spd    T26BUF         014027'
C21            000100  spd    MA15           004216'        MLAST   400000 000000  spd    T26M           007305'
C22            000101  spd    MA15PN         004220'        MPRUN          000010  spd    T27BUF         014030'
C23            000102  spd    MA16           004527'        MSG     100000 000000  spd    T27M           007613'
C24            000103  spd    MA16PN         004531'        NDCB    000200 000000  spd    T2M            000717'
C3             000056  spd    MA17           005040'        NDMP    000400 000000  spd    T30BUF         014031'
C4             000057  spd    MA17PN         005042'        ODELAY         012521' ext    T30M           010127'
C5             000060  spd    MA2            000715'        P              000017         T31BUF         014032'
C8             000063  spd    MA20           005352'        PAT            000014         T31M           010445'
CALL    200000 000000  spd    MA20PN         005354'        PATBUF         001674' ext    T32BUF         014033'
CBUF           013210' ext    MA21           005632'        PDEC    006000 000000         T32M           010763'
CBUFSZ         000315  spd    MA21PN         005634'        PNTDEC  037640 000000         T33BUF         014034'
CBUS    100000 000000  spd    MA22           006111'        PNTHW   037540 000000         T33M           011247'
CCALL          000005  spd    MA22PN         006113'        PNTMSG  037000 000000         T34BUF         014035'
CCALLC         000006  spd    MA23           006367'        PORTCI         001710' ext    T34M           011532'
CCOMP          000010  spd    MA23PN         006371'        PORTNI         001706' ext    T35BUF         014036'
CCPNT          000000  ext    MA24           006650'        RDCSR          007031' ext    T35M           012014'
CDERR          002643' ext    MA24PN         006652'        RDEBUF         012470' ext    T36BUF         014037'
CEBUFA         013105' ext    MA25           007027'        RTN     263740 000000         T36M           012275'
CEXEC          002571' ext    MA25P1         007031'        SAVCSR         014010'        T37M           012536'
CEXIT          000012  spd    MA25PN         007043'        SAVDA1         014011'        T3M            001123'
CHINIT         012424' ext    MA26           007272'        SAVDA2         014012'        T4M            001220'
CJRST          000011  spd    MA26PN         007274'        SAVDAT         014013'        T5M            001453'
CLAST          000000  spd    MA27           007600'        SAVNUM         014014'        T6M            002022'
CLEN           013102' ext    MA27PN         007602'        SCOPER  027000 000000         T6MM           001717'
CPAT           002641' ext    MA3            001121'        SCOSW          000000  ext    T7M            002334'
CRAMPE  004000 000000  spd    MA30           010114'        SDATA          006777' ext    TA1            000022'
CSETRD         000001  spd    MA30PN         010116'        SNEXT          006775' ext    TA10           002570'
CSETRW         000003  spd    MA31           010432'        SSLAST         000000  spd    TA11           003056'
CSETWR         000002  spd    MA31PN         010434'        SSSTRT         000001  spd    TA12           003117'
CSRPNT         007037' ext    MA32           010750'        T10M           002645'        TA13           003262'
CSTART         000004  spd    MA32PN         010752'        T11M           003074'        TA14           003563'
CSTATF         001715' ext    MA33           011234'        T12M           003175'        TA15           004063'
CWAIT          000007  spd    MA33PN         011236'        T13BUF         014015'        TA16           004376'
E1             000001  spd    MA34           011517'        T13M           003427'        TA17           004707'
E4             000004  spd    MA34PN         011521'        T14BUF         014016'        TA2            000657'
EBUSPE         004000  spd    MA35           012001'        T14M           003726'        TA20           005217'
ERFLG          000015         MA35PN         012003'        T15BUF         014017'        TA21           005530'
GENCCW         012430' ext    MA36           012262'        T15M           004231'        TA22           006011'
GO      260740 000000         MA36PN         012264'        T16BUF         014020'        TA23           006267'
```

```
DFPTA9  MACRO %53B(1242) 09:42  1-Jul-85 Page S-2
DFPTA9  MAC     18-Apr-85 11:00          SYMBOL TABLE                                                     SEQ 1809

TA24            006546'         TG36            012117'         TS14            003574'         TS22            006022'
TA25            006771'         TG37            012377'         TS14A           003576'         TS22A           006024'
TA26            007135'         TG4             001162'         TS14IN          003650'         TS22IN          006052'
TA27            007446'         TG5             001305'         TS14L           003575'         TS22L           006023'
TA3             001066'         TG6             001621'         TS14NO          003660'         TS22NO          006062'
TA30            007762'         TG7             002225'         TS14NE          003653'         TS22NE          006055'
TA31            010300'         TJRST           000006  spd     TS14NM          014041'         TS22NM          014047'
TA32            010616'         TL1             000016'         TS14NN          003676'         TS22NN          006101'
TA33            011133'         TL10            002533'         TS14RS          003701'         TS22RS          006104'
TA34            011416'         TL11            003052'         TS15            004074'         TS23            006300'
TA35            011700'         TL13            003225'         TS15A           004076'         TS23A           006302'
TA36            012161'         TL14            003526'         TS15IN          004150'         TS23IN          006330'
TA37            012443'         TL15            004026'         TS15L           004075'         TS23L           006301'
TA4             001174'         TL16            004341'         TS15NO          004160'         TS23NO          006340'
TA5             001316'         TL17            004652'         TS15NE          004153'         TS23NE          006333'
TA6             001632'         TL2             000653'         TS15NM          014042'         TS23NM          014050'
TA7             002267'         TL20            005162'         TS15NN          004201'         TS23NN          006357'
TB10            002571'         TL21            005473'         TS15RS          004204'         TS23RS          006362'
TB3             001067'         TL22            005754'         TS16            004407'         TS24            006557'
TB4             001175'         TL23            006232'         TS16A           004411'         TS24A           006561'
TB5             001317'         TL24            006511'         TS16IN          004463'         TS24IN          006607'
TB6             001633'         TL26            007100'         TS16L           004410'         TS24L           006560'
TB7             002270'         TL27            007411'         TS16NO          004473'         TS24NO          006617'
TCALL           000003  spd     TL3             001062'         TS16NE          004466'         TS24NE          006612'
TCALLC          000004  spd     TL30            007725'         TS16NM          014043'         TS24NM          014051'
TCHECK          000005  spd     TL31            010243'         TS16NN          004512'         TS24NN          006640'
TEBUFA          012471' ext     TL32            010561'         TS16RS          004515'         TS24RS          006643'
TEBUFC          012460' ext     TL33            011076'         TS17            004720'         TS26            007146'
TEXEC           012443' ext     TL34            011361'         TS17A           004722'         TS26A           007150'
TEXIT           000007  spd     TL35            011643'         TS17IN          004774'         TS26IN          007222'
TG1             000011'         TL36            012124'         TS17L           004721'         TS26L           007147'
TG10            002526'         TL37            012404'         TS17NO          005004'         TS26NO          007233'
TG11            003045'         TL4             001167'         TS17NE          004777'         TS26NE          007225'
TG12            003112'         TL5             001312'         TS17NM          014044'         TS26NM          014052'
TG13            003220'         TL6             001626'         TS17NN          005023'         TS26NN          007255'
TG14            003521'         TL7             002232'         TS17RS          005026'         TS26RS          007260'
TG15            004021'         TLAST           000000  spd     TS2             000670'         TS27            007457'
TG16            004334'         TLOAD           012402' ext     TS20            005230'         TS27A           007461'
TG17            004645'         TRACE           012400' ext     TS20A           005232'         TS27IN          007533'
TG2             000646'         TS1             000033'         TS20IN          005304'         TS27L           007460'
TG20            005155'         TS10            002602'         TS20L           005231'         TS27NO          007543'
TG21            005466'         TS10A           002603'         TS20NO          005314'         TS27NE          007536'
TG22            005747'         TS10IN          002616'         TS20NE          005307'         TS27NM          014053'
TG23            006225'         TS10SE          002613'         TS20NM          014045'         TS27NN          007563'
TG24            006504'         TS10SW          002623'         TS20NN          005335'         TS27RS          007566'
TG25            006764'         TS11            003067'         TS20RS          005340'         TS3             001100'
TG26            007073'         TS13            003273'         TS21            005541'         TS30            007773'
TG27            007404'         TS13A           003275'         TS21A           005543'         TS30A           007775'
TG3             001055'         TS13IN          003347'         TS21IN          005571'         TS30IN          010047'
TG30            007720'         TS13L           003274'         TS21L           005542'         TS30L           007774'
TG31            010236'         TS13NO          003357'         TS21NO          005601'         TS30NO          010057'
TG32            010554'         TS13NE          003352'         TS21NE          005574'         TS30NE          010052'
TG33            011071'         TS13NM          014040'         TS21NM          014046'         TS30NM          014054'
TG34            011354'         TS13NN          003377'         TS21NN          005622'         TS30NN          010077'
TG35            011636'         TS13RS          003402'         TS21RS          005625'         TS30RS          010102'
```

```
DFPTA9  MACRO %53B(1242) 09:42  1-Jul-85 Page S-3
DFPTA9  MAC      18-Apr-85 11:00          SYMBOL TABLE                                                   SEQ 1810

TS31      010311'   TS37A     012500'       TSTC4           001152' ent   .LAND   000000 spd
TS31A     010313'   TS37CH    012510'       TSTC40          000000  ext   .LB     000000 spd
TS31IN    010365'   TS37DA    014063'       TSTC41          000000  ext   .LBAD   000000 spd
TS31L     010312'   TS37IN    012506'       TSTC5           001275' ent   .LCCCA  000000 spd
TS31NO    010375'   TS3A      001101'       TSTC6           001612' ent   .LCCCE  000000 spd
TS31NE    010370'   TS3EBU    014064'       TSTC7           002215' ent   .LCCCP  000000 spd
TS31NM    014055'   TS3INC    001112'       TSTEBF          200000  spd   .LCCER  000000 spd
TS31NN    010415'   TS3SEE    001116'       TSTSUB          012406' ext   .LCCFZ  000000 spd
TS31RS    010420'   TS3SET    001110'       TTPNT           000000  ext   .LCCGC  000000 spd
TS32      010627'   TS4       001206'       TX1             000032'       .LCCLW  000000 spd
TS32A     010631'   TS5       001330'       TX10            002601'       .LCCMB  000000 spd
TS32IN    010703'   TS5A      001331'       TX11            003066'       .LCENA  000000 spd
TS32L     010630'   TS5CHE    001356'       TX12            003154'       .LCJP   000000 spd
TS32NO    010713'   TS5INC    001346'       TX13            003272'       .LCJS   000000 spd
TS32NE    010706'   TS5SET    001340'       TX14            003573'       .LCONT  000000 spd
TS32NM    014056'   TS6       001644'       TX15            004073'       .LCRTN  000000 spd
TS32NN    010733'   TS6A      001645'       TX16            004406'       .LCRY   000000 spd
TS32RS    010736'   TS6CHE    001730'       TX17            004717'       .LD     000000 spd
TS33      011144'   TS6INC    001666'       TX2             000667'       .LJ     010000 spd
TS33A     011146'   TS6S      001647'       TX20            005227'       .LJMAP  000000 spd
TS33IN    011174'   TS6SET    001654'       TX21            005540'       .LJZ    000000 spd
TS33L     011145'   TS7       002301'       TX22            006021'       .LLDCT  000000 spd
TS33NO    011204'   TS7A      002302'       TX23            006277'       .LMGC   000001 spd
TS33NE    011177'   TS7IN     002314'       TX24            006556'       .LOENA  002000 spd
TS33NM    014057'   TS7SE     002311'       TX25            007026'       .LOR    000000 spd
TS33NN    011224'   TSLOD1    001707' ext   TX26            007145'       .LPLUS  000000 spd
TS33RS    011227'   TSLOD2    001711' ext   TX27            007456'       .LRPCT  000000 spd
TS34      011427'   TSSTAR    000010  spd   TX3             001077'       .LSOA   000000 spd
TS34A     011431'   TSTC1     000000' ent   TX30            007772'       .LSOB   000000 spd
TS34IN    011457'   TSTC10    002517' ent   TX31            010310'       .LSAB   000000 spd
TS34L     011430'   TSTC11    003034' ent   TX32            010626'       .LSDO   000000 spd
TS34NO    011467'   TSTC12    003101' ent   TX33            011143'       .LSELC  000000 spd
TS34NE    011462'   TSTC13    003210' ent   TX34            011426'       .LSELE  000000 spd
TS34NM    014060'   TSTC14    003510' ent   TX35            011710'       .LSELF  000000 spd
TS34NN    011507'   TSTC15    004007' ent   TX36            012171'       .LSELM  000000 spd
TS34RS    011512'   TSTC16    004322' ent   TX37            012476'       .LSELP  000000 spd
TS35      011711'   TSTC17    004633' ent   TX4             001205'       .LSKCN  000000 spd
TS35A     011713'   TSTC2     000637' ent   TX5             001327'       .LSMIN  000000 spd
TS35IN    011741'   TSTC20    005144' ent   TX6             001643'       .LXNOR  000000 spd
TS35L     011712'   TSTC21    005456' ent   TX7             002300'       .LXOR   000000 spd
TS35NO    011751'   TSTC22    005736' ent   TXALL   060000  000000  spd   .MA     000017 spd
TS35NE    011744'   TSTC23    006215' ent   TXNOT   040000  000000  spd   .MAND   000007 spd
TS35NM    014061'   TSTC24    006473' ent   UDEBUG          002546' ext   .MB     000017 spd
TS35NN    011771'   TSTC25    006754' ent   USER            030037        .MBAD   000001 spd
TS35RS    011774'   TSTC26    007062' ent   Z9              000000'       .MCCCA  000037 spd
TS36      012172'   TSTC27    007372' ent   ZCBUS   000004  000000  spd   .MCCCE  000037 spd
TS36A     012174'   TSTC3     001044' ent   $ARG2           000004        .MCCCP  000037 spd
TS36IN    012222'   TSTC30    007710' ent   $B              000043        .MCCER  000037 spd
TS36L     012173'   TSTC31    010224' ent   $CHR            554743        .MCCFZ  000037 spd
TS36NO    012232'   TSTC32    010542' ent   $GARG           000001        .MCCGC  000037 spd
TS36NE    012225'   TSTC33    011060' ent   %ADDR           005211  spd   .MCCLW  000037 spd
TS36NM    014062'   TSTC34    011344' ent   %ML     521052  102004  spd   .MCCMB  000037 spd
TS36NN    012252'   TSTC35    011627' ent   %MR     431040  005040  spd   .MCENA  000001 spd
TS36RS    012255'   TSTC36    012111' ent   .LA             000000  spd   .MCJP   000017 spd
TS37      012477'   TSTC37    012372' ent   .LADDR  000100  000000  spd   .MCJS   000017 spd
```

```
DFPTA9  MACRO %53B(1242) 09:42  1-Jul-85 Page S-4
DFPTA9  MAC       18-Apr-85 11:00          SYMBOL TABLE

.MCONT          000017  spd      .RSOB   300000  000000  spd
.MCRTN          000017  spd      .RSAB   100000  000000  spd
.MCRY           000001  spd      .RSDO   700000  000000  spd
.MD             000007  spd      .RSELC          004000  spd
.MJ             007777  spd      .RSELE          005000  spd
.MJMAP          000017  spd      .RSELF          003000  spd
.MJZ            000017  spd      .RSELM          002000  spd
.MLDCT          000017  spd      .RSELP          001000  spd
.MMGC           001777  spd      .RSKCN          240000  spd
.MOENA          000001  spd      .RSMIN  010000  000000  spd
.MOR            000007  spd      .RXNOR  070000  000000  spd
.MPLUS          000007  spd      .RXOR   060000  000000  spd
.MRPCT          000017  spd
.MSOA           000007  spd
.MSOB           000007  spd
.MSAB           000007  spd
.MSDO           000007  spd
.MSELC          000007  spd
.MSELE          000007  spd
.MSELF          000007  spd
.MSELM          000007  spd
.MSELP          000007  spd
.MSKCN          000037  spd
.MSMIN          000007  spd
.MXNOR          000007  spd
.MXOR           000007  spd
.RA     000010  000000  spd
.RAND   040000  000000  spd
.RB             400000  spd
.RBAD           000001  spd
.RCCCA          [illegible]  spd
.RCCCE          160000  spd
.RCCCP          140000  spd
.RCCER          100000  spd
.RCCFZ          020000  spd
.RCCGC          010000  spd
.RCCLW          170000  spd
.RCCMB          120000  spd
.RCENA  000400  000000  spd
.RCJP           000060  spd
.RCJS           000020  spd
.RCONT          000340  spd
.RCRTN          000240  spd
.RCRY           000400  spd
.RD     001000  000000  spd
.RJ             000000  spd
.RJMAP          000040  spd
.RJZ            000000  spd
.RLDCT          000300  spd
.RMGC           000000  spd
.ROENA          000000  spd
.ROR    030000  000000  spd
.RPLUS          000000  spd
.RRPCT          000220  spd
.RSOA   400000  000000  spd
```

```
BBPNT      22#    190    713   2481
BEXEC      22#     91    670   2456
BUFCOM     32#   1661   1668   1969   2263
BUFF       21#    927   1891   1895   1898   2174   2178   2181   2244   2245   2247   2248   2249   2666
          2667   2683   2799   2806   2978   2979   2995   3111   3118   3292   3293   3309   3425   3432
          3617   3618   3634   3750   3757   3940   3941   3957   4073   4080   4262   4263   4279   4395
          4402   4585   4586   4602   4683   4690   4874   4875   4891   4972   4979   5160   5161   5177
          5258   5265   5447   5448   5464   5545   5552   5847   5848   5864   5980   5987   6170   6171
          6187   6303   6310   6496   6497   6513   6629   6636   6824   6825   6841   6957   6964   7152
          7153   7169   7285   7292   7479   7480   7496   7577   7584   7770   7771   7787   7868   7875
          8060   8061   8077   8158   8165   8349   8350   8366   8447   8454   8635   8636   8654   8750
BUFGEN     32#
C1         861   1003   1172   1493   1859   2143
C12         67    648   1172   1493   1859   2143   2432   2518   2641   2952   3265   3590   3913   4236
          4560   4848   5135   5421   5699   5821   6143   6471   6797   7125   7453   7745   8036   8326
          8613
C13        648
C14        861   1172   1859
C15       1003   1493   2143
C16       1493   2143
C17       1172   1859
C18         67   2432
C19        648
C2          67    648    861   1003   1172   1493   1859   2143   2432   2518   2641   2952   3265   3590
          3913   4236   4560   4848   5135   5421   5699   5821   6143   6471   6797   7125   7453   7745
          8036   8326   8613
C20         67    648   2432   2518   2641   2952   3265   3590   3913   4236   4560   4848   5135   5421
          5699   5821   6143   6471   6797   7125   7453   7745   8036   8326   8613
C21        648
C22        648
C23         67    648   2432
C24         67    648   2432
C3          67    648   2432
C4          67    648    861   1003   1172   1493   1859   2143   2432   2518   2641   2952   3265   3590
          3913   4236   4560   4848   5135   5421   5699   5821   6143   6471   6797   7125   7453   7745
          8036   8326   8613
C5         861   1003   1172   1493   1859   2143
C8          67    648    861   1003   1172   1493   1859   2143   2432   2518   2641   2952   3265   3590
          3913   4236   4560   4848   5135   5421   5699   5821   6143   6471   6797   7125   7453   7745
          8036   8326   8613
CALL       190    713    934   1058   1261   1262   1612   1613   1961   1962   2255   2256   2481   2573
          2585   2840   2841   3150   3151   3467   3468   3790   3791   4113   4114   4437   4438   4725
          4726   5012   5013   5298   5299   5587   5588   5753   5765   6024   6025   6344   6345   6670
          6671   6998   6999   7326   7327   7618   7619   7909   7910   8199   8200   8488   8489   8745
          8746
CBUF       21#   1896   2179   2681   2993   3307   3632   3955   4277   4600   4889   5175   5462   5862
          6185   6511   6839   7167   7494   7785   8075   8364   8652
CBUFSZ    1896   2179   2681   2993   3307   3632   3955   4277   4600   4889   5175   5462   5862   6185
          6511   6839   7167   7494   7785   8075   8364   8652
CBUS        65    646    859   1001   1170   1491   1857   2141   2430   2516   2639   2950   3263   3588
          3911   4234   4558   4846   5133   5419   5697   5819   6141   6469   6795   7123   7451   7743
          8034   8324   8611
CCALL      904    905    907    908   1214   1215   1219   1220   1534   1535   1539   1540   1935   1936
```

```
            2218    2219    2221    2222
CCALLC       905     906    1215    1216    1535    1536    1936    1937    2219    2220
CCOMP       1940    1941    2224    2225
CCPNT        23#     934    1058    1262    1613    1962    2256
CDERR        23#    1669    1964    1970    2258    2264
CEBUFA       23#    1248    1267    1276    1280    1316    1599    1618    1626
CEXEC        23#     886    1028    1196    1516    1917    2200
CEXIT       1049
CHINIT       28#    1897    2180    2682    2994    3308    3633    3956    4278    4601    4890    5176    5463    5863
            6186    6512    6840    7168    7495    7786    8076    8365    8653
CJRST        910     911    1220    1221    1540    1541    1941    1942    2225    2226
CLAST        911     912    1053    1054    1221    1222    1541    1542    1942    1943    2226    2227
CLEN         23#    1229    1249    1268    1273    1549    1600    1619    1623
CPAT         23#     915     920    1225    1234    1545    1559    1947    1952    1968    2231    2236    2262
CRAMPE      2558    5738
CSETRD      1046    1047    1050    1051    1536    1537
CSETRW      1937    1938    2220    2221
CSETWR       906     907    1216    1217
CSRPNT       21#    2581    5761
CSTART       908     909    1047    1048    1051    1052    1217    1218    1537    1538    1938    1939    2222    2223
CSTATF       23#     881    1022    1242    1583
CWAIT        909     910    1048    1049    1052    1053    1218    1219    1538    1539    1939    1940    2223    2224
E1          1859    2143
E4          1859    2143
EBUSPE      2560    5740
ERFLG         84      94     101     663     673     680     878     889     896    1019    1023    1031    1038    1188
            1191    1199    1206    1250    1253    1266    1312    1508    1511    1519    1526    1601    1604    1617
            1645    1663    1875    1878    1920    1927    2158    2161    2203    2210    2449    2459    2466    2535
            2543    2544    2545    2549    2557    2559    2657    2706    2713    2969    3018    3025    3283    3332
            3339    3608    3657    3664    3931    3980    3987    4253    4302    4309    4576    4625    4632    4865
            4914    4921    5151    5200    5207    5438    5487    5494    5715    5723    5724    5725    5729    5737
            5739    5838    5887    5894    6161    6210    6217    6487    6536    6543    6815    6864    6871    7143
            7192    7199    7470    7519    7526    7761    7810    7817    8051    8100    8107    8340    8389    8396
            8626    8677    8706
GENCCW       28#    1901    2184    2686    2998    3312    3637    3960    4282    4605    4894    5180    5467    5867
            6190    6516    6844    7172    7499    7790    8080    8369    8657
IPACLR       27#      85     664     885    1027    1195    1235    1515    1558    1882    1905    1916    1953    2165
            2188    2199    2237    2450    2538    2658    2970    3284    3609    3932    4254    4577    4866    5152
            5439    5718    5839    6162    6488    6816    7144    7471    7762    8052    8341    8627    8681
IPASRT       27#    2542    5722
LAST         190     713     934    1058    1262    1613    1962    2256    2481    2573    2585    2841    3151    3468
            3791    4114    4438    4726    5013    5299    5588    5753    5765    6025    6345    6671    6999    7327
            7619    7910    8200    8489    8746
LDCSR        27#    1288    1293    1295    1299    1304    1306    1632    1637    1639    1887    1910    2170    2193
            2561    2675    2677    2695    2699    2820    2825    2827    2987    2989    3007    3011    3135    3137
            3301    3303    3321    3325    3452    3454    3626    3628    3646    3650    3775    3777    3949    3951
            3969    3973    4098    4100    4271    4273    4291    4295    4422    4424    4594    4596    4614    4618
            4710    4712    4883    4885    4903    4907    4997    4999    5169    5171    5189    5193    5283    5285
            5456    5458    5476    5480    5572    5574    5741    5856    5858    5876    5880    6004    6009    6011
            6179    6181    6199    6203    6329    6331    6505    6507    6525    6529    6655    6657    6833    6835
            6853    6857    6983    6985    7161    7163    7181    7185    7311    7313    7488    7490    7508    7512
            7603    7605    7779    7781    7799    7803    7894    7896    8069    8071    8089    8093    8184    8186
            8358    8360    8378    8382    8473    8475    8644    8648    8666    8670    8686    8688    8695    8697
```

```
             8736     8740
LDRAR         27#     1291     1302     1635     1885     1908     2168     2191     2673     2693     2823     2985     3005     3133
             3299     3319     3450     3624     3644     3773     3947     3967     4096     4269     4289     4420     4592     4612
             4708     4881     4901     4995     5167     5187     5281     5454     5474     5570     5854     5874     6007     6177
             6197     6327     6503     6523     6653     6831     6851     6981     7159     7179     7309     7486     7506     7601
             7777     7797     7892     8067     8087     8182     8356     8376     8471     8642     8664     8684     8693     8734
MA1            98      189#
MA10         2207     2254#
MA10CH       2255     2258#
MA11         2463     2480#
MA12         2546     2572#
MA12P1       2573     2575#
MA12PN       2585     2587#
MA13         2710     2840#
MA13PN       2840     2843#
MA14         3022     3150#
MA14PN       3150     3153#
MA15         3336     3467#
MA15PN       3467     3470#
MA16         3661     3790#
MA16PN       3790     3793#
MA17         3984     4113#
MA17PN       4113     4116#
MA2           677      712#
MA20         4306     4437#
MA20PN       4437     4440#
MA21         4629     4725#
MA21PN       4725     4728#
MA22         4918     5012#
MA22PN       5012     5015#
MA23         5204     5298#
MA23PN       5298     5301#
MA24         5491     5587#
MA24PN       5587     5590#
MA25         5726     5752#
MA25P1       5753     5755#
MA25PN       5765     5767#
MA26         5891     6024#
MA26PN       6024     6027#
MA27         6214     6344#
MA27PN       6344     6347#
MA3           893      933#
MA30         6540     6670#
MA30PN       6670     6673#
MA31         6868     6998#
MA31PN       6998     7001#
MA32         7196     7326#
MA32PN       7326     7329#
MA33         7523     7618#
MA33PN       7618     7621#
MA34         7814     7909#
MA34PN       7909     7912#
MA35         8104     8199#
```

```
MA35PN     8199    8202#
MA36       8393    8488#
MA36PN     8488    8491#
MA37       8703    8745#
MA37PN     8745    8748#
MA4        1035    1057#
MA5        1203    1260#
MA5CHO     1266    1269    1278    1287#
MA5CHE     1247    1261    1266#
MA6        1523    1611#
MA6CHO     1617    1620    1631#
MA6CH1     1647    1659#
MA6CHE     1598    1612    1617#
MA7        1924    1960#
MA7CHE     1961    1964#
MB12       2564    2584#
MB25       5744    5764#
MBCN       1576    1578
MBUS         65
MLAST        67     648     861    1003    1172    1493    1859    2143    2432    2518    2641    2952    3265    3590
           3913    4236    4560    4848    5135    5421    5699    5821    6143    6471    6797    7125    7453    7745
           8036    8326    8613
MPRUN      1292    1303    1636    1886    1909    2169    2192    2674    2694    2824    2986    3006    3134    3300
           3320    3451    3625    3645    3774    3948    3968    4097    4270    4290    4421    4593    4613    4709
           4882    4902    4996    5168    5188    5282    5455    5475    5571    5855    5875    6008    6178    6198
           6328    6504    6524    6654    6832    6852    6982    7160    7180    7310    7487    7507    7602    7778
           7798    7893    8068    8088    8183    8357    8377    8472    8643    8665    8685    8694    8735
MSG         189     712     933    1057    1260    1611    1960    2254    2480    2572    2584    5752    5764
NDCB         65     646     859    1001    1170    1491    1857    2141    2430    2516    2639    2950    3263    3588
           3911    4234    4558    4846    5133    5419    5697    5819    6141    6469    6795    7123    7451    7743
           8034    8324    8611
NDMP         65     646     859    1001    1170    1491    1857    2141    2430    2516    2639    2950    3263    3588
           3911    4234    4558    4846    5133    5419    5697    5819    6141    6469    6795    7123    7451    7743
           8034    8324    8611
ODELAY       21#   1912    2195    2555    2697    3009    3323    3648    3971    4293    4616    4905    5191    5478
           5735    5878    6201    6527    6855    7183    7510    7801    8091    8380    8646    8668    8738
P           922    1237    1240    1254    1255    1561    1564    1605    1606    1955    2239    2807    3119    3433
           3758    4081    4403    4691    4980    5266    5553    5988    6311    6637    6965    7293    7585    7876
           8166    8455    8728
PAT        1968    2262
PATBUF       32#   1566
PORTCI       22#   1578
PORTNI       22#   1576
RDCSR        27#   2556    2575    5736    5755
RDEBUF       27#   1296    1307    1640    2562    5742    8689    8698
SAVCSR     2577#   2577    2578    5757#   5757    5758
SAVDA1     1251    1297#   1297    1313    1324    1567    1602    1646    1650
SAVDA2     1252    1308#   1308    1321    1603    1641    1653
SAVDAT     2563#   2563    2588    5743#   5743    5768
SAVNUM     1662#   1662
SCOSW        32#
SDATA        27#   2541    5721
SNEXT        27#   2539    5719
```

```
SSLAST      185    186    708    709   2476   2477
SSSTRT      109    110    111    112    113    114    115    116    117    118    119    120    121    122
            123    124    125    126    127    128    129    130    131    132    133    134    135    136
            137    138    139    140    141    142    143    144    145    146    147    148    149    150
            151    152    153    154    155    156    157    158    159    160    161    162    163    164
            165    166    167    168    169    170    171    172    173    174    175    176    177    178
            179    180    181    182    183    184    185    688    689    690    691    692    693    694
            695    696    697    698    699    700    701    702    703    704    705    706    707    708
           2474   2475   2476
T10M       2142   2152   2269#
T11M       2431   2443   2485#
T12M       2517   2529   2594#
T13BUF     2800#  2800   2805   2844
T13M       2640   2651   2855#
T14BUF     3112#  3112   3117   3154
T14M       2951   2963   3165#
T15BUF     3426#  3426   3431   3471
T15M       3264   3277   3482#
T16BUF     3751#  3751   3756   3794
T16M       3589   3602   3805#
T17BUF     4074#  4074   4079   4117
T17M       3912   3925   4128#
T1M          66     78    194#
T20BUF     4396#  4396   4401   4441
T20M       4235   4247   4452#
T21BUF     4684#  4684   4689   4729
T21M       4559   4570   4740#
T22BUF     4973#  4973   4978   5016
T22M       4847   4859   5027#
T23BUF     5259#  5259   5264   5302
T23M       5134   5145   5313#
T24BUF     5546#  5546   5551   5591
T24M       5420   5432   5602#
T25M       5698   5709   5774#
T26BUF     5981#  5981   5986   6028
T26M       5820   5832   6039#
T27BUF     6304#  6304   6309   6348
T27M       6142   6155   6359#
T2M         647    657    717#
T30BUF     6630#  6630   6635   6674
T30M       6470   6481   6685#
T31BUF     6958#  6958   6963   7002
T31M       6796   6809   7013#
T32BUF     7286#  7286   7291   7330
T32M       7124   7137   7341#
T33BUF     7578#  7578   7583   7622
T33M       7452   7464   7633#
T34BUF     7869#  7869   7874   7913
T34M       7744   7755   7924#
T35BUF     8159#  8159   8164   8203
T35M       8035   8045   8214#
T36BUF     8448#  8448   8453   8492
T36M       8325   8334   8503#
```

```
T37M           8612       8620       8758#
T3M             860        872        938#
T4M            1002       1013       1062#
T5M            1171       1182       1330#
T6M            1492       1502       1676#
T6MM           1569       1571       1573       1575       1580       1586#
T7M            1858       1869       1975#
TA1              91#        93        101
TA10           2199#      2210
TA11           2456#      2458       2466
TA12           2535#      2547       2565
TA13           2703#      2705       2713
TA14           3015#      3017       3025
TA15           3329#      3331       3339
TA16           3654#      3656       3664
TA17           3977#      3979       3987
TA2             670#       672        680
TA20           4299#      4301       4309
TA21           4622#      4624       4632
TA22           4911#      4913       4921
TA23           5197#      5199       5207
TA24           5484#      5486       5494
TA25           5715#      5727       5745
TA26           5884#      5886       5894
TA27           6207#      6209       6217
TA3             885#       896
TA30           6533#      6535       6543
TA31           6861#      6863       6871
TA32           7189#      7191       7199
TA33           7516#      7518       7526
TA34           7807#      7809       7817
TA35           8097#      8099       8107
TA36           8386#      8388       8396
TA37           8674#      8676       8706
TA4            1027#      1038
TA5            1195#      1206
TA6            1515#      1526
TA7            1916#      1927
TB10           2200#      2202
TB3             886#       888
TB4            1028#      1030
TB5            1196#      1198
TB6            1516#      1518
TB7            1917#      1919
TCALL          2721       2722       2726       2727       2733       2734       2740       2741       2747       2748       2755       2756       2762       2763
               2769       2770       2776       2777       2783       2784       2790       2791       3033       3034       3038       3039       3045       3046
               3052       3053       3059       3060       3067       3068       3074       3075       3081       3082       3088       3089       3095       3096
               3102       3103       3347       3348       3352       3353       3359       3360       3366       3367       3373       3374       3381       3382
               3388       3389       3395       3396       3402       3403       3409       3410       3416       3417       3672       3673       3677       3678
               3684       3685       3691       3692       3698       3699       3706       3707       3713       3714       3720       3721       3727       3728
               3734       3735       3741       3742       3995       3996       4000       4001       4007       4008       4014       4015       4021       4022
               4029       4030       4036       4037       4043       4044       4050       4051       4057       4058       4064       4065       4317       4318
               4322       4323       4329       4330       4336       4337       4343       4344       4351       4352       4358       4359       4365       4366
```

```
          4372  4373  4379  4380  4386  4387  4640  4641  4645  4646  4652  4653  4660  4661
          4667  4668  4674  4675  4929  4930  4934  4935  4941  4942  4948  4949  4956  4957
          4963  4964  5215  5216  5220  5221  5227  5228  5234  5235  5241  5242  5249  5250
          5502  5503  5507  5508  5514  5515  5521  5522  5528  5529  5536  5537  5902  5903
          5907  5908  5914  5915  5921  5922  5928  5929  5936  5937  5943  5944  5950  5951
          5957  5958  5964  5965  5971  5972  6225  6226  6230  6231  6237  6238  6244  6245
          6251  6252  6258  6259  6266  6267  6273  6274  6280  6281  6287  6288  6294  6295
          6551  6552  6556  6557  6563  6564  6570  6571  6577  6578  6585  6586  6592  6593
          6599  6600  6606  6607  6613  6614  6620  6621  6879  6880  6884  6885  6891  6892
          6898  6899  6905  6906  6913  6914  6920  6921  6927  6928  6934  6935  6941  6942
          6948  6949  7207  7208  7212  7213  7219  7220  7226  7227  7233  7234  7240  7241
          7248  7249  7255  7256  7262  7263  7269  7270  7276  7277  7534  7535  7539  7540
          7546  7547  7553  7554  7560  7561  7568  7569  7825  7826  7830  7831  7837  7838
          7844  7845  7851  7852  7858  7859  8115  8116  8120  8121  8127  8128  8134  8135
          8141  8142  8149  8150  8404  8405  8409  8410  8416  8417  8423  8424  8430  8431
          8438  8439  8714  8715
TCALLC    2722  2723  3034  3035  3348  3349  3673  3674  3996  3997  4318  4319  4641  4642
          4930  4931  5216  5217  5503  5504  5903  5904  6226  6227  6552  6553  6880  6881
          7208  7209  7535  7536  7826  7827  8116  8117  8405  8406  8715  8716
TCHECK    2728  2729  2735  2736  2742  2743  2749  2750  2757  2758  2764  2765  2771  2772
          2778  2779  2785  2786  2792  2793  3040  3041  3047  3048  3054  3055  3061  3062
          3069  3070  3076  3077  3083  3084  3090  3091  3097  3098  3104  3105  3354  3355
          3361  3362  3368  3369  3375  3376  3383  3384  3390  3391  3397  3398  3404  3405
          3411  3412  3418  3419  3679  3680  3686  3687  3693  3694  3700  3701  3708  3709
          3715  3716  3722  3723  3729  3730  3736  3737  3743  3744  4002  4003  4009  4010
          4016  4017  4023  4024  4031  4032  4038  4039  4045  4046  4052  4053  4059  4060
          4066  4067  4324  4325  4331  4332  4338  4339  4345  4346  4353  4354  4360  4361
          4367  4368  4374  4375  4381  4382  4388  4389  4647  4648  4654  4655  4662  4663
          4669  4670  4676  4677  4936  4937  4943  4944  4950  4951  4958  4959  4965  4966
          5222  5223  5229  5230  5236  5237  5243  5244  5251  5252  5509  5510  5516  5517
          5523  5524  5530  5531  5538  5539  5909  5910  5916  5917  5923  5924  5930  5931
          5938  5939  5945  5946  5952  5953  5959  5960  5966  5967  5973  5974  6232  6233
          6239  6240  6246  6247  6253  6254  6260  6261  6268  6269  6275  6276  6282  6283
          6289  6290  6296  6297  6558  6559  6565  6566  6572  6573  6579  6580  6587  6588
          6594  6595  6601  6602  6608  6609  6615  6616  6622  6623  6886  6887  6893  6894
          6900  6901  6907  6908  6915  6916  6922  6923  6929  6930  6936  6937  6943  6944
          6950  6951  7214  7215  7221  7222  7228  7229  7235  7236  7242  7243  7250  7251
          7257  7258  7264  7265  7271  7272  7278  7279  7541  7542  7548  7549  7555  7556
          7562  7563  7570  7571  7832  7833  7839  7840  7846  7847  7853  7854  7860  7861
          8122  8123  8129  8130  8136  8137  8143  8144  8151  8152  8411  8412  8418  8419
          8425  8426  8432  8433  8440  8441  8717  8718
TEBUFA     22#  8699
TEBUFC     22#  8690
TEXEC      22#  2703  3015  3329  3654  3977  4299  4622  4911  5197  5484  5884  6207  6533
          6861  7189  7516  7807  8097  8386  8674
TEXIT     2729  2730  2736  2737  2743  2744  2750  2751  2758  2759  2765  2766  2772  2773
          2779  2780  2786  2787  2793  2794  3041  3042  3048  3049  3055  3056  3062  3063
          3070  3071  3077  3078  3084  3085  3091  3092  3098  3099  3105  3106  3355  3356
          3362  3363  3369  3370  3376  3377  3384  3385  3391  3392  3398  3399  3405  3406
          3412  3413  3419  3420  3680  3681  3687  3688  3694  3695  3701  3702  3709  3710
          3716  3717  3723  3724  3730  3731  3737  3738  3744  3745  4003  4004  4010  4011
          4017  4018  4024  4025  4032  4033  4039  4040  4046  4047  4053  4054  4060  4061
          4067  4068  4325  4326  4332  4333  4339  4340  4346  4347  4354  4355  4361  4362
          4368      4369      4375      4376      4382      4383      4389      4390      4648      4649      4655      4656      4663      4664
          4670      4671      4677      4678      4937      4938      4944      4945      4951      4952      4959      4960      4966      4967
          5223      5224      5230      5231      5237      5238      5244      5245      5252      5253      5510      5511      5517      5518
          5524      5525      5531      5532      5539      5540      5910      5911      5917      5918      5924      5925      5931      5932
          5939      5940      5946      5947      5953      5954      5960      5961      5967      5968      5974      5975      6233      6234
          6240      6241      6247      6248      6254      6255      6261      6262      6269      6270      6276      6277      6283      6284
          6290      6291      6297      6298      6559      6560      6566      6567      6573      6574      6580      6581      6588      6589
          6595      6596      6602      6603      6609      6610      6616      6617      6623      6624      6887      6888      6894      6895
          6901      6902      6908      6909      6916      6917      6923      6924      6930      6931      6937      6938      6944      6945
          6951      6952      7215      7216      7222      7223      7229      7230      7236      7237      7243      7244      7251      7252
          7258      7259      7265      7266      7272      7273      7279      7280      7542      7543      7549      7550      7556      7557
          7563      7564      7571      7572      7833      7834      7840      7841      7847      7848      7854      7855      7861      7862
          8123      8124      8130      8131      8137      8138      8144      8145      8152      8153      8412      8413      8419      8420
          8426      8427      8433      8434      8441      8442      8718      8719
TG1         64        76#
TG10      2140      2150#
TG11      2429      2441#
TG12      2515      2527#
TG13      2638      2649#
TG14      2949      2961#
TG15      3262      3275#
TG16      3587      3600#
TG17      3910      3923#
TG2        645       655#
TG20      4233      4245#
TG21      4557      4568#
TG22      4845      4857#
TG23      5132      5143#
TG24      5418      5430#
TG25      5696      5707#
TG26      5818      5830#
TG27      6140      6153#
TG3        858       870#
TG30      6468      6479#
TG31      6794      6807#
TG32      7122      7135#
TG33      7450      7462#
TG34      7742      7753#
TG35      8033      8043#
TG36      8323      8332#
TG37      8610      8618#
TG4       1000      1011#
TG5       1169      1180#
TG6       1490      1500#
TG7       1856      1867#
TJRST     2794      2795      3106      3107      3420      3421      3745      3746      4068      4069      4390      4391      4678      4679
          4967      4968      5253      5254      5540      5541      5975      5976      6298      6299      6624      6625      6952      6953
          7280      7281      7572      7573      7862      7863      8153      8154      8442      8443      8719      8720
TL1         84#       99
TL10      2158#     2208
TL11      2449#     2464
TL13      2657#     2711
TL14      2969#     3023
```

```
TL15         3283#      3337
TL16         3608#      3662
TL17         3931#      3985
TL2           663#       678
TL20         4253#      4307
TL21         4576#      4630
TL22         4865#      4919
TL23         5151#      5205
TL24         5438#      5492
TL26         5838#      5892
TL27         6161#      6215
TL3           878#       894
TL30         6487#      6541
TL31         6815#      6869
TL32         7143#      7197
TL33         7470#      7524
TL34         7761#      7815
TL35         8051#      8105
TL36         8340#      8394
TL37         8626#      8704
TL4          1019#      1036
TL5          1188#      1204
TL6          1508#      1524
TL7          1875#      1925
TLAST        2795       2796       3107       3108       3421       3422       3746       3747       4069       4070       4391       4392       4679       4680
             4968       4969       5254       5255       5541       5542       5976       5977       6299       6300       6625       6626       6953       6954
             7281       7282       7573       7574       7863       7864       8154       8155       8443       8444       8720       8721
TLOAD          21#        79        658        873       1014       1183       1503       1581       1870       2153       2444       2530       2652       2964
             3278       3603       3926       4248       4571       4860       5146       5433       5710       5833       6156       6482       6810       7138
             7465       7756       8046       8335       8621
TRACE          21#        77        656        871       1012       1181       1501       1868       2151       2442       2528       2650       2962       3276
             3601       3924       4246       4569       4858       5144       5431       5708       5831       6154       6480       6808       7136       7463
             7754       8044       8333       8619
TS1            87        109#
TS10         2159       2218#
TS10A        2219#      2225
TS10IN       2219       2236#
TS10SE       2218       2230#
TS10SW       2221       2244#
TS11         2452       2474#
TS13         2687       2721#
TS13A        2726#
TS13IN       2721       2799#
TS13L        2722#      2794
TS13NO       2810#      2814
TS13NE       2722       2805#
TS13NM       2815#      2815       2833       2848
TS13NN       2726       2733       2740       2747       2755       2762       2769       2776       2783       2790       2832#
TS13RS       2728       2735       2742       2749       2757       2764       2771       2778       2785       2792       2809       2813       2836#
TS14         2999       3033#
TS14A        3038#
TS14IN       3033       3111#
TS14L        3034#      3106
```

```
TS14NO     3122#   3126
TS14NE     3034    3117#
TS14NM     3127#   3127    3143    3158
TS14NN     3038    3045    3052    3059    3067    3074    3081    3088    3095    3102    3142#
TS14RS     3040    3047    3054    3061    3069    3076    3083    3090    3097    3104    3121    3125    3146#
TS15       3313    3347#
TS15A      3352#
TS15IN     3347    3425#
TS15L      3348#   3420
TS15NO     3436#   3443
TS15NE     3348    3431#
TS15NM     3444#   3444    3460    3475
TS15NN     3352    3359    3366    3373    3381    3388    3395    3402    3409    3416    3459#
TS15RS     3354    3361    3368    3375    3383    3390    3397    3404    3411    3418    3435    3437    3442    3463#
TS16       3638    3672#
TS16A      3677#
TS16IN     3672    3750#
TS16L      3673#   3745
TS16NO     3761#   3766
TS16NE     3673    3756#
TS16NM     3767#   3767    3783    3798
TS16NN     3677    3684    3691    3698    3706    3713    3720    3727    3734    3741    3782#
TS16RS     3679    3686    3693    3700    3708    3715    3722    3729    3736    3743    3760    3765    3786#
TS17       3961    3995#
TS17A      4000#
TS17IN     3995    4073#
TS17L      3996#   4068
TS17NO     4084#   4089
TS17NE     3996    4079#
TS17NM     4090#   4090    4106    4121
TS17NN     4000    4007    4014    4021    4029    4036    4043    4050    4057    4064    4105#
TS17RS     4002    4009    4016    4023    4031    4038    4045    4052    4059    4066    4083    4088    4109#
TS2         666     688#
TS20       4283    4317#
TS20A      4322#
TS20IN     4317    4395#
TS20L      4318#   4390
TS20NO     4406#   4413
TS20NE     4318    4401#
TS20NM     4414#   4414    4430    4445
TS20NN     4322    4329    4336    4343    4351    4358    4365    4372    4379    4386    4429#
TS20RS     4324    4331    4338    4345    4353    4360    4367    4374    4381    4388    4405    4407    4412    4433#
TS21       4606    4640#
TS21A      4645#
TS21IN     4640    4683#
TS21L      4641#   4678
TS21NO     4694#   4701
TS21NE     4641    4689#
TS21NM     4702#   4702    4718    4733
TS21NN     4645    4652    4660    4667    4674    4717#
TS21RS     4647    4654    4662    4669    4676    4693    4695    4700    4721#
TS22       4895    4929#
TS22A      4934#
```

```
TS22IN      4929    4972#
TS22L       4930#   4967
TS22NO      4983#   4988
TS22NE      4930    4978#
TS22NM      4989#   4989    5005    5020
TS22NN      4934    4941    4948    4956    4963    5004#
TS22RS      4936    4943    4950    4958    4965    4982    4987    5008#
TS23        5181    5215#
TS23A       5220#
TS23IN      5215    5258#
TS23L       5216#   5253
TS23NO      5269#   5274
TS23NE      5216    5264#
TS23NM      5275#   5275    5291    5306
TS23NN      5220    5227    5234    5241    5249    5290#
TS23RS      5222    5229    5236    5243    5251    5268    5273    5294#
TS24        5468    5502#
TS24A       5507#
TS24IN      5502    5545#
TS24L       5503#   5540
TS24NO      5556#   5563
TS24NE      5503    5551#
TS24NM      5564#   5564    5580    5595
TS24NN      5507    5514    5521    5528    5536    5579#
TS24RS      5509    5516    5523    5530    5538    5555    5557    5562    5583#
TS26        5868    5902#
TS26A       5907#
TS26IN      5902    5980#
TS26L       5903#   5975
TS26NO      5992#   5998
TS26NE      5903    5986#
TS26NM      5999#   5999    6017    6032
TS26NN      5907    5914    5921    5928    5936    5943    5950    5957    5964    5971    6016#
TS26RS      5909    5916    5923    5930    5938    5945    5952    5959    5966    5973    5991    5997    6020#
TS27        6191    6225#
TS27A       6230#
TS27IN      6225    6303#
TS27L       6226#   6298
TS27NO      6314#   6320
TS27NE      6226    6309#
TS27NM      6321#   6321    6337    6352
TS27NN      6230    6237    6244    6251    6258    6266    6273    6280    6287    6294    6336#
TS27RS      6232    6239    6246    6253    6260    6268    6275    6282    6289    6296    6313    6315    6319    6340#
TS3          879     904#
TS30        6517    6551#
TS30A       6556#
TS30IN      6551    6629#
TS30L       6552#   6624
TS30NO      6640#   6646
TS30NE      6552    6635#
TS30NM      6647#   6647    6663    6678
TS30NN      6556    6563    6570    6577    6585    6592    6599    6606    6613    6620    6662#
TS30RS      6558    6565    6572    6579    6587    6594    6601    6608    6615    6622    6639    6641    6645    6666#
```

```
TS31        6845      6879#
TS31A       6884#
TS31IN      6879      6957#
TS31L       6880#     6952
TS31NO      6968#     6974
TS31NE      6880      6963#
TS31NM      6975#     6975      6991      7006
TS31NN      6884      6891      6898      6905      6913      6920      6927      6934      6941      6948      6990#
TS31RS      6886      6893      6900      6907      6915      6922      6929      6936      6943      6950      6967      6969      6973      6994#
TS32        7173      7207#
TS32A       7212#
TS32IN      7207      7285#
TS32L       7208#     7280
TS32NO      7296#     7302
TS32NE      7208      7291#
TS32NM      7303#     7303      7319      7334
TS32NN      7212      7219      7226      7233      7240      7248      7255      7262      7269      7276      7318#
TS32RS      7214      7221      7228      7235      7242      7250      7257      7264      7271      7278      7295      7297      7301      7322#
TS33        7500      7534#
TS33A       7539#
TS33IN      7534      7577#
TS33L       7535#     7572
TS33NO      7588#     7594
TS33NE      7535      7583#
TS33NM      7595#     7595      7611      7626
TS33NN      7539      7546      7553      7560      7568      7610#
TS33RS      7541      7548      7555      7562      7570      7587      7589      7593      7614#
TS34        7791      7825#
TS34A       7830#
TS34IN      7825      7868#
TS34L       7826#     7862
TS34NO      7879#     7885
TS34NE      7826      7874#
TS34NM      7886#     7886      7902      7917
TS34NN      7830      7837      7844      7851      7858      7901#
TS34RS      7832      7839      7846      7853      7860      7878      7880      7884      7905#
TS35        8081      8115#
TS35A       8120#
TS35IN      8115      8158#
TS35L       8116#     8153
TS35NO      8169#     8175
TS35NE      8116      8164#
TS35NM      8176#     8176      8192      8207
TS35NN      8120      8127      8134      8141      8149      8191#
TS35RS      8122      8129      8136      8143      8151      8168      8170      8174      8195#
TS36        8370      8404#
TS36A       8409#
TS36IN      8404      8447#
TS36L       8405#     8442
TS36NO      8458#     8464
TS36NE      8405      8453#
TS36NM      8465#     8465      8481      8496
TS36NN      8409      8416      8423      8430      8438      8480#
```

```
TS36RS    8411     8418     8425     8432     8440     8457     8459     8463     8484#
TS37      8658     8714#
TS37A     8715#    8719
TS37CH    8715     8726#
TS37DA    8723#    8723     8726     8749
TS37IN    8714     8723#
TS3A       905#     910
TS3EBU     909      928#     928
TS3INC     905      920#
TS3SEE     907      927#
TS3SET     904      915#
TS4       1020     1046#
TS5       1189     1214#
TS5A      1215#    1220
TS5CHE    1219     1247#
TS5INC    1215     1234#
TS5SET    1214     1225#
TS6       1509     1534#
TS6A      1535#    1540
TS6CHE    1539     1598#
TS6INC    1535     1558#
TS6S      1537#    1553
TS6SET    1534     1545#
TS7       1876     1935#
TS7A      1936#    1941
TS7IN     1936     1952#
TS7SE     1935     1946#
TSLOD1      21#    1577
TSLOD2      21#    1579
TSSTAR    2727     2734     2741     2748     2756     2763     2770     2777     2784     2791     3039     3046     3053     3060
          3068     3075     3082     3089     3096     3103     3353     3360     3367     3374     3382     3389     3396     3403
          3410     3417     3678     3685     3692     3699     3707     3714     3721     3728     3735     3742     4001     4008
          4015     4022     4030     4037     4044     4051     4058     4065     4323     4330     4337     4344     4352     4359
          4366     4373     4380     4387     4646     4653     4661     4668     4675     4935     4942     4949     4957     4964
          5221     5228     5235     5242     5250     5508     5515     5522     5529     5537     5908     5915     5922     5929
          5937     5944     5951     5958     5965     5972     6231     6238     6245     6252     6259     6267     6274     6281
          6288     6295     6557     6564     6571     6578     6586     6593     6600     6607     6614     6621     6885     6892
          6899     6906     6914     6921     6928     6935     6942     6949     7213     7220     7227     7234     7241     7249
          7256     7263     7270     7277     7540     7547     7554     7561     7569     7831     7838     7845     7852     7859
          8121     8128     8135     8142     8150     8410     8417     8424     8431     8439     8716
TSTC1       10       64#
TSTC10      10     1860     2140#
TSTC11      11       69      649     2429#
TSTC12      11     2515#
TSTC13      11     2519     2638#
TSTC14      11     2520     2642     2949#
TSTC15      11     2953     3262#
TSTC16      11     2954     3266     3587#
TSTC17      11     3267     3591     3910#
TSTC2       10       68      645#
TSTC20      11     3268     3592     3914     4233#
TSTC21      12     3269     3593     3915     4237     4557#
TSTC22      12     3270     3594     3916     4238     4561     4845#
```

```
TSTC23        12       3595       3917       4239       4562       4849       5132#
TSTC24        12       3918       4240       4563       4850       5136       5418#
TS1C25        12       2521       2643       2955       4851       5137       5422       5696#
TSTC26        12       2522       2644       2956       4852       5138       5423       5700       5818#
TSTC27        12       5701       5822       6140#
TSTC3         10         70        858#      2433
TSTC30        12       5702       5823       6144       6468#
TSTC31        13       5424       5824       6145       6472       6794#
TSTC32        13       5425       5825       6146       6473       6798       7122#
TSTC33        13       6147       6474       6799       7126       7450#
TSTC34        13       6148       6800       7127       7454       7742#
TSTC35        13       6801       7128       7455       7746       8033#
TSTC36        13       6802       7129       7456       7747       8037       8323#
TSTC37        13       7130       7457       7748       8038       8327       8610#
TSTC4         10         71        862       1000#      2434
TSTC40        17#       650        864       1005       1173       1494       1861       2144
TSTC41        17#       865       1006       1174       1495       1862       2145
TSTC5         10        863       1004       1169#      2435
TSTC6         10       1175       1490#      2436
TSTC7         10       1856#
TSTEBF      1294       1305       1638       2560       5740       8687       8696
TSTSUB        21#        86        665        880       1021       1190       1510       1877       2160       2451       2537       2553       2659       2971
            3285       3610       3933       4255       4578       4867       5153       5440       5717       5733       5840       6163       6489       6817
            7145       7472       7763       8053       8342       8628
TTPNT         22#      2841       3151       3468       3791       4114       4438       4726       5013       5299       5588       6025       6345       6671
            6999       7327       7619       7910       8200       8489       8746
TX1           92        100        105#
TX10        2201       2209       2214#
TX11        2457       2465       2470#
TX12        2548       2549       2566       2568#
TX13        2704       2712       2717#
TX14        3016       3024       3029#
TX15        3330       3338       3343#
TX16        3655       3663       3668#
TX17        3978       3986       3991#
TX2          671        679        684#
TX20        4300       4308       4313#
TX21        4623       4631       4636#
TX22        4912       4920       4925#
TX23        5198       5206       5211#
TX24        5485       5493       5498#
TX25        5728       5729       5746       5748#
TX26        5885       5893       5898#
TX27        6208       6216       6221#
TX3          887        895        900#
TX30        6534       6542       6547#
TX31        6862       6870       6875#
TX32        7190       7198       7203#
TX33        7517       7525       7530#
TX34        7808       7816       7821#
TX35        8098       8106       8111#
TX36        8387       8395       8400#
TX37        8675       8705       8710#
```

```
TX4         1029     1037     1042#
TX5         1197     1205     1210#
TX6         1517     1525     1530#
TX7         1918     1926     1931#
TXALL        190      713      934     1058     1262     1613     1962     2256     2481     2573     2585     2841     3151     3468
            3791     4114     4438     4726     5013     5299     5588     5753     5765     6025     6345     6671     6999     7327
            7619     7910     8200     8489     8746
TXNOT        189      712      933     1057     1260     1261     1611     1612     1960     1961     2254     2255     2480     2572
            2584     2840     3150     3467     3790     4113     4437     4725     5012     5298     5587     5752     5764     6024
            6344     6670     6998     7326     7618     7909     8199     8488     8745
UDEBUG        21#      22#    1238     1562     1892     2175
USER        1227     1547     1551
Z9            39#      76      655      870     1011     1180     1500     1867     2150     2441     2527     2649     2961     3275
            3600     3923     4245     4568     4857     5143     5430     5707     5830     6153     6479     6807     7135     7462
            7753     8043     8332     8618
ZCBUS         65      646      859     1001     1170     1491     1857     2141     2430     2516     2639     2950     3263     3588
            3911     4234     4558     4846     5133     5419     5697     5819     6141     6469     6795     7123     7451     7743
            8034     8324     8611
$ARG2        194#     194      196#     196      198#     198      200#     200      202#     202      204#     204      206#     206
             208#     208      210#     210      212#     212      214#     214      216#     216      218#     218      220#     220
             222#     222      224#     224      226#     226      228#     228      230#     230      232#     232      234#     234
             236#     236      238#     238      240#     240      242#     242      244#     244      246#     246      248#     248
             250#     250      252#     252      254#     254      256#     256      258#     258      260#     260      262#     262
             264#     264      266#     266      268#     268      273#     273      275#     275      280#     280      282#     282
             287#     287      289#     289      291#     291      296#     296      298#     298      300#     300      305#     305
             307#     307      309#     309      314#     314      316#     316      318#     318      323#     323      325#     325
             327#     327      332#     332      334#     334      336#     336      341#     341      343#     343      345#     345
             350#     350      352#     352      354#     354      359#     359      361#     361      363#     363      368#     368
             370#     370      372#     372      377#     377      379#     379      381#     381      386#     386      388#     388
             390#     390      395#     395      397#     397      399#     399      404#     404      406#     406      408#     408
             413#     413      415#     415      417#     417      422#     422      424#     424      426#     426      431#     431
             433#     433      435#     435      440#     440      442#     442      444#     444      449#     449      451#     451
             453#     453      458#     458      460#     460      462#     462      467#     467      469#     469      471#     471
             476#     476      478#     478      480#     480      485#     485      487#     487      489#     489      494#     494
             496#     496      498#     498      503#     503      505#     505      507#     507      512#     512      514#     514
             516#     516      521#     521      523#     523      525#     525      530#     530      532#     532      534#     534
             539#     539      541#     541      543#     543      548#     548      550#     550      552#     552      557#     557
             559#     559      561#     561      566#     566      568#     568      570#     570      575#     575      577#     577
             579#     579      584#     584      586#     586      588#     588      593#     593      595#     595      597#     597
             599#     599      604#     604      609#     609      611#     611      613#     613      615#     615      617#     617
             717#     717      719#     719      721#     721      723#     723      725#     725      727#     727      729#     729
             731#     731      733#     733      735#     735      740#     740      742#     742      747#     747      749#     749
             754#     754      756#     756      758#     758      763#     763      765#     765      767#     767      772#     772
             774#     774      776#     776      781#     781      783#     783      785#     785      790#     790      792#     792
             794#     794      799#     799      801#     801      803#     803      808#     808      810#     810      812#     812
             817#     817      822#     822      824#     824      826#     826      828#     828      830#     830      832#     832
             938#     938      943#     943      948#     948      950#     950      955#     955      957#     957      959#     959
             964#     964      966#     966      971#     971      973#     973     1062#    1062     1064#    1064     1066#    1066
            1071#    1071     1076#    1076     1081#    1081     1083#    1083     1088#    1088     1093#    1093     1095#    1095
            1100#    1100     1102#    1102     1108#    1108     1110#    1110     1112#    1112     1114#    1114     1116#    1116
            1118#    1118     1120#    1120     1122#    1122     1127#    1127     1129#    1129     1330#    1330     1332#    1332
            1334#    1334     1336#    1336     1341#    1341     1343#    1343     1345#    1345     1347#    1347     1349#    1349
            1351#    1351     1356#    1356     1358#    1358     1360#    1360     1365#    1365     1371#    1371     1373#    1373
```

EQ 1814

SEQ 1827

```
1375#  1375   1377#  1377   1379#  1379   1381#  1381   1383#  1383   1385#  1385   1392#  1392
1394#  1394   1396#  1396   1398#  1398   1400#  1400   1402#  1402   1404#  1404   1406#  1406
1408#  1408   1410#  1410   1412#  1412   1414#  1414   1416#  1416   1421#  1421   1423#  1423
1428#  1428   1430#  1430   1432#  1432   1437#  1437   1439#  1439   1444#  1444   1446#  1446
1451#  1451   1456#  1456   1458#  1458   1586#  1586   1588#  1588   1590#  1590   1592#  1592
1676#  1676   1678#  1678   1680#  1680   1682#  1682   1684#  1684   1686#  1686   1691#  1691
1693#  1693   1695#  1695   1697#  1697   1699#  1699   1701#  1701   1706#  1706   1708#  1708
1710#  1710   1712#  1712   1714#  1714   1716#  1716   1718#  1718   1720#  1720   1722#  1722
1724#  1724   1726#  1726   1728#  1728   1730#  1730   1732#  1732   1734#  1734   1736#  1736
1738#  1738   1740#  1740   1742#  1742   1744#  1744   1749#  1749   1751#  1751   1753#  1753
1755#  1755   1757#  1757   1759#  1759   1761#  1761   1763#  1763   1765#  1765   1767#  1767
1773#  1773   1775#  1775   1777#  1777   1783#  1783   1785#  1785   1787#  1787   1789#  1789
1791#  1791   1793#  1793   1795#  1795   1797#  1797   1802#  1802   1804#  1804   1809#  1809
1811#  1811   1813#  1813   1818#  1818   1823#  1823   1825#  1825   1975#  1975   1977#  1977
1979#  1979   1981#  1981   1984#  1984   1986#  1986   1988#  1988   1990#  1990   1992#  1992
1994#  1994   1997#  1997   2002#  2002   2008#  2008   2010#  2010   2012#  2012   2014#  2014
2016#  2016   2018#  2018   2020#  2020   2022#  2022   2027#  2027   2029#  2029   2031#  2031
2033#  2033   2035#  2035   2037#  2037   2039#  2039   2041#  2041   2043#  2043   2045#  2045
2047#  2047   2049#  2049   2051#  2051   2053#  2053   2055#  2055   2057#  2057   2059#  2059
2061#  2061   2063#  2063   2065#  2065   2070#  2070   2072#  2072   2074#  2074   2080#  2080
2082#  2082   2084#  2084   2086#  2086   2088#  2088   2090#  2090   2092#  2092   2094#  2094
2096#  2096   2098#  2098   2100#  2100   2102#  2102   2107#  2107   2109#  2109   2269#  2269
2272#  2272   2274#  2274   2276#  2276   2278#  2278   2280#  2280   2283#  2283   2285#  2285
2287#  2287   2289#  2289   2292#  2292   2298#  2298   2300#  2300   2302#  2302   2308#  2308
2310#  2310   2312#  2312   2314#  2314   2316#  2316   2318#  2318   2320#  2320   2322#  2322
2327#  2327   2329#  2329   2331#  2331   2333#  2333   2335#  2335   2337#  2337   2339#  2339
2341#  2341   2343#  2343   2345#  2345   2347#  2347   2349#  2349   2351#  2351   2353#  2353
2355#  2355   2357#  2357   2359#  2359   2361#  2361   2363#  2363   2365#  2365   2370#  2370
2372#  2372   2374#  2374   2380#  2380   2382#  2382   2384#  2384   2386#  2386   2388#  2388
2390#  2390   2392#  2392   2394#  2394   2396#  2396   2398#  2398   2400#  2400   2402#  2402
2407#  2407   2409#  2409   2485#  2485   2487#  2487   2594#  2594   2596#  2596   2598#  2598
2600#  2600   2602#  2602   2855#  2855   2857#  2857   2859#  2859   2861#  2861   2866#  2866
2868#  2868   2870#  2870   2872#  2872   2874#  2874   2876#  2876   2881#  2881   2883#  2883
2885#  2885   2890#  2890   2892#  2892   2897#  2897   2899#  2899   2901#  2901   2903#  2903
2905#  2905   2907#  2907   2909#  2909   2911#  2911   2913#  2913   3165#  3165   3167#  3167
3169#  3169   3171#  3171   3176#  3176   3178#  3178   3180#  3180   3182#  3182   3184#  3184
3186#  3186   3191#  3191   3193#  3193   3195#  3195   3200#  3200   3202#  3202   3207#  3207
3209#  3209   3211#  3211   3213#  3213   3215#  3215   3217#  3217   3219#  3219   3221#  3221
3223#  3223   3482#  3482   3484#  3484   3486#  3486   3488#  3488   3493#  3493   3495#  3495
3497#  3497   3499#  3499   3501#  3501   3503#  3503   3508#  3508   3510#  3510   3512#  3512
3514#  3514   3516#  3516   3518#  3518   3520#  3520   3525#  3525   3527#  3527   3532#  3532
3534#  3534   3536#  3536   3538#  3538   3540#  3540   3542#  3542   3544#  3544   3546#  3546
3548#  3548   3805#  3805   3807#  3807   3809#  3809   3811#  3811   3816#  3816   3818#  3818
3820#  3820   3822#  3822   3824#  3824   3826#  3826   3831#  3831   3833#  3833   3835#  3835
3837#  3837   3839#  3839   3841#  3841   3843#  3843   3848#  3848   3650#  3850   3855#  3855
3857#  3857   3859#  3859   3861#  3861   3863#  3863   3865#  3865   3867#  3867   3869#  3869
3871#  3871   4128#  4128   4130#  4130   4132#  4132   4134#  4134   4139#  4139   4141#  4141
4143#  4143   4145#  4145   4147#  4147   4149#  4149   4154#  4154   4156#  4156   4158#  4158
4160#  4160   4162#  4162   4164#  4164   4166#  4166   4171#  4171   4173#  4173   4178#  4178
4180#  4180   4182#  4182   4184#  4184   4186#  4186   4188#  4188   4190#  4190   4192#  4192
4194#  4194   4452#  4452   4454#  4454   4456#  4456   4458#  4458   4463#  4463   4465#  4465
4467#  4467   4469#  4469   4471#  4471   4473#  4473   4478#  4478   4480#  4480   4482#  4482
4484#  4484   4486#  4486   4488#  4488   4490#  4490   4495#  4495   4497#  4497   4502#  4502
```

```
4504#   4504    4506#   4506    4508#   4508    4510#   4510    4512#   4512    4514#   4514    4516#   4516
4518#   4518    4740#   4740    4742#   4742    4744#   4744    4746#   4746    4751#   4751    4753#   4753
4755#   4755    4757#   4757    4759#   4759    4761#   4761    4766#   4766    4768#   4768    4770#   4770
4772#   4772    4774#   4774    4776#   4776    4778#   4778    4783#   4783    4785#   4785    4790#   4790
4792#   4792    4794#   4794    4796#   4796    4798#   4798    4800#   4800    4802#   4802    4804#   4804
4806#   4806    5027#   5027    5029#   5029    5031#   5031    5033#   5033    5038#   5038    5040#   5040
5042#   5042    5044#   5044    5046#   5046    5048#   5048    5053#   5053    5055#   5055    5057#   5057
5059#   5059    5061#   5061    5063#   5063    5065#   5065    5070#   5070    5072#   5072    5077#   5077
5079#   5079    5081#   5081    5083#   5083    5085#   5085    5087#   5087    5089#   5089    5091#   5091
5093#   5093    5313#   5313    5315#   5315    5317#   5317    5319#   5319    5324#   5324    5326#   5326
5328#   5328    5330#   5330    5332#   5332    5334#   5334    5339#   5339    5341#   5341    5343#   5343
5345#   5345    5347#   5347    5349#   5349    5351#   5351    5356#   5356    5358#   5358    5363#   5363
5365#   5365    5367#   5367    5369#   5369    5371#   5371    5373#   5373    5375#   5375    5377#   5377
5379#   5379    5602#   5602    5604#   5604    5606#   5606    5608#   5608    5613#   5613    5615#   5615
5617#   5617    5619#   5619    5621#   5621    5623#   5623    5628#   5628    5630#   5630    5632#   5632
5634#   5634    5636#   5636    5638#   5638    5640#   5640    5645#   5645    5647#   5647    5652#   5652
5654#   5654    5656#   5656    5658#   5658    5660#   5660    5662#   5662    5664#   5664    5666#   5666
5668#   5668    5774#   5774    5776#   5776    5778#   5778    5780#   5780    5782#   5782    6039#   6039
6041#   6041    6043#   6043    6045#   6045    6050#   6050    6052#   6052    6054#   6054    6056#   6056
6058#   6058    6060#   6060    6062#   6062    6064#   6064    6069#   6069    6071#   6071    6073#   6073
6078#   6078    6080#   6080    6085#   6085    6087#   6087    6089#   6089    6091#   6091    6093#   6093
6095#   6095    6097#   6097    6099#   6099    6101#   6101    6359#   6359    6361#   6361    6363#   6363
6365#   6365    6370#   6370    6372#   6372    6374#   6374    6376#   6376    6378#   6378    6380#   6380
6382#   6382    6384#   6384    6389#   6389    6391#   6391    6393#   6393    6395#   6395    6397#   6397
6399#   6399    6401#   6401    6406#   6406    6408#   6408    6413#   6413    6415#   6415    6417#   6417
6419#   6419    6421#   6421    6423#   6423    6425#   6425    6427#   6427    6429#   6429    6685#   6685
6687#   6687    6689#   6689    6691#   6691    6696#   6696    6698#   6698    6700#   6700    6702#   6702
6704#   6704    6706#   6706    6708#   6708    6710#   6710    6715#   6715    6717#   6717    6719#   6719
6721#   6721    6723#   6723    6725#   6725    6727#   6727    6732#   6732    6734#   6734    6739#   6739
6741#   6741    6743#   6743    6745#   6745    6747#   6747    6749#   6749    6751#   6751    6753#   6753
6755#   6755    7013#   7013    7015#   7015    7017#   7017    7019#   7019    7024#   7024    7026#   7026
7028#   7028    7030#   7030    7032#   7032    7034#   7034    7036#   7036    7038#   7038    7043#   7043
7045#   7045    7047#   7047    7049#   7049    7051#   7051    7053#   7053    7055#   7055    7060#   7060
7062#   7062    7067#   7067    7069#   7069    7071#   7071    7073#   7073    7075#   7075    7077#   7077
7079#   7079    7081#   7081    7083#   7083    7341#   7341    7343#   7343    7345#   7345    7347#   7347
7352#   7352    7354#   7354    7356#   7356    7358#   7358    7360#   7360    7362#   7362    7364#   7364
7366#   7366    7371#   7371    7373#   7373    7375#   7375    7377#   7377    7379#   7379    7381#   7381
7383#   7383    7388#   7388    7390#   7390    7395#   7395    7397#   7397    7399#   7399    7401#   7401
7403#   7403    7405#   7405    7407#   7407    7409#   7409    7411#   7411    7633#   7633    7635#   7635
7637#   7637    7639#   7639    7644#   7644    7646#   7646    7648#   7648    7650#   7650    7652#   7652
7654#   7654    7656#   7656    7658#   7658    7663#   7663    7665#   7665    7667#   7667    7669#   7669
7671#   7671    7673#   7673    7675#   7675    7680#   7680    7682#   7682    7687#   7687    7689#   7689
7691#   7691    7693#   7693    7695#   7695    7697#   7697    7699#   7699    7701#   7701    7703#   7703
7924#   7924    7926#   7926    7928#   7928    7930#   7930    7935#   7935    7937#   7937    7939#   7939
7941#   7941    7943#   7943    7945#   7945    7947#   7947    7949#   7949    7954#   7954    7956#   7956
7958#   7958    7960#   7960    7962#   7962    7964#   7964    7966#   7966    7971#   7971    7973#   7973
7978#   7978    7980#   7980    7982#   7982    7984#   7984    7986#   7986    7988#   7988    7990#   7990
7992#   7992    7994#   7994    8214#   8214    8216#   8216    8218#   8218    8220#   8220    8225#   8225
8227#   8227    8229#   8229    8231#   8231    8233#   8233    8235#   8235    8237#   8237    8239#   8239
8244#   8244    8246#   8246    8248#   8248    8250#   8250    8252#   8252    8254#   8254    8256#   8256
8261#   8261    8263#   8263    8268#   8268    8270#   8270    8272#   8272    8274#   8274    8276#   8276
8278#   8278    8280#   8280    8282#   8282    8284#   8284    8503#   8503    8505#   8505    8507#   8507
8509#   8509    8514#   8514    8516#   8516    8518#   8518    8520#   8520    8522#   8522    8524#   8524
```

```
EQ 1816      8526#  8526   8528#  8528   8533#  8533   8535#  8535   8537#  8537   8539#  8539   8541#  8541    SEQ 1829
             8543#  8543   8545#  8545   8550#  8550   8552#  8552   8557#  8557   8559#  8559   8561#  8561
             8563#  8563   8565#  8565   8567#  8567   8569#  8569   8571#  8571   8573#  8573   8758#  8758
             8760#  8760   8762#  8762   8764#  8764   8766#  8766   8768#  8768   8770#  8770   8772#  8772
             8774#  8774   8776#  8776   8778#  8778   8780#  8780   8782#  8782   8784#  8784   8786#  8786
             8791#  8791   8793#  8793   8795#  8795   8797#  8797   8799#  8799   8801#  8801   8803#  8803
             8805#  8805   8810#  8810   8812#  8812   8814#  8814   8819#  8819   8821#  8821   8823#  8823
             8825#  8825   8830#  8830   8832#  8832   8837#  8837   8839#  8839   8841#  8841   8843#  8843
             8845#  8845   8847#  8847   8849#  8849   8851#  8851   8853#  8853   8858#  8858   8860#  8860
             8862#  8862   8864#  8864   8866#  8866   8868#  8868   8870#  8870   8872#  8872   8874#  8874
  SB          194#   194    196#   196    198#   198    200#   200    202#   202    204#   204    206#   206
              208#   208    210#   210    212#   212    214#   214    216#   216    218#   218    220#   220
              222#   222    224#   224    226#   226    228#   228    230#   230    232#   232    234#   234
              236#   236    238#   238    240#   240    242#   242    244#   244    246#   246    248#   248
              250#   250    252#   252    254#   254    256#   256    258#   258    260#   260    262#   262
              264#   264    266#   266    268#   268    273#   273    275#   275    280#   280    282#   282
              287#   287    289#   289    291#   291    296#   296    298#   298    300#   300    305#   305
              307#   307    309#   309    314#   314    316#   316    318#   318    323#   323    325#   325
              327#   327    332#   332    334#   334    336#   336    341#   341    343#   343    345#   345
              350#   350    352#   352    354#   354    359#   359    361#   361    363#   363    368#   368
              370#   370    372#   372    377#   377    379#   379    381#   381    386#   386    388#   388
              390#   390    395#   395    397#   397    399#   399    404#   404    406#   406    408#   408
              413#   413    415#   415    417#   417    422#   422    424#   424    426#   426    431#   431
              433#   433    435#   435    440#   440    442#   442    444#   444    449#   449    451#   451
              453#   453    458#   458    460#   460    462#   462    467#   467    469#   469    471#   471
              476#   476    478#   478    480#   480    485#   485    487#   487    489#   489    494#   494
              496#   496    498#   498    503#   503    505#   505    507#   507    512#   512    514#   514
              516#   516    521#   521    523#   523    525#   525    530#   530    532#   532    534#   534
              539#   539    541#   541    543#   543    548#   548    550#   550    552#   552    557#   557
              559#   559    561#   561    566#   566    568#   568    570#   570    575#   575    577#   577
              579#   579    584#   584    586#   586    588#   588    593#   593    595#   595    597#   597
              599#   599    604#   604    609#   609    611#   611    613#   613    615#   615    617#   617
              717#   717    719#   719    721#   721    723#   723    725#   725    727#   727    729#   729
              731#   731    733#   733    735#   735    740#   740    742#   742    747#   747    749#   749
              754#   754    756#   756    758#   758    763#   763    765#   765    767#   767    772#   772
              774#   774    776#   776    781#   781    783#   783    785#   785    790#   790    792#   792
              794#   794    799#   799    801#   801    803#   803    808#   808    810#   810    812#   812
              817#   817    822#   822    824#   824    826#   826    828#   828    830#   830    832#   832
              938#   938    943#   943    948#   948    950#   950    955#   955    957#   957    959#   959
              964#   964    966#   966    971#   971    973#   973   1062#  1062   1064#  1064   1066#  1066
             1071#  1071   1076#  1076   1081#  1081   1083#  1083   1088#  1088   1093#  1093   1095#  1095
             1100#  1100   1102#  1102   1108#  1108   1110#  1110   1112#  1112   1114#  1114   1116#  1116
             1118#  1118   1120#  1120   1122#  1122   1127#  1127   1129#  1129   1330#  1330   1332#  1332
             1334#  1334   1336#  1336   1341#  1341   1343#  1343   1345#  1345   1347#  1347   1349#  1349
             1351#  1351   1356#  1356   1358#  1358   1360#  1360   1365#  1365   1371#  1371   1373#  1373
             1375#  1375   1377#  1377   1379#  1379   1381#  1381   1383#  1383   1385#  1385   1392#  1392
             1394#  1394   1396#  1396   1398#  1398   1400#  1400   1402#  1402   1404#  1404   1406#  1406
             1408#  1408   1410#  1410   1412#  1412   1414#  1414   1416#  1416   1421#  1421   1423#  1423
             1428#  1428   1430#  1430   1432#  1432   1437#  1437   1439#  1439   1444#  1444   1446#  1446
             1451#  1451   1456#  1456   1458#  1458   1586#  1586   1588#  1588   1590#  1590   1592#  1592
             1676#  1676   1678#  1678   1680#  1680   1682#  1682   1684#  1684   1686#  1685   1691#  1691
             1693#  1693   1695#  1695   1697#  1697   1699#  1699   1701#  1701   1706#  1706   1708#  1708
             1710#  1710   1712#  1712   1714#  1714   1716#  1716   1718#  1718   1720#  1720   1722#  1722
```

```
EQ 1817  1724#  1724   1726#  1726   1728#  1728   1730#  1730   1732#  1732   1734#  1734   1736#  1736   SEQ 1830
         1738#  1738   1740#  1740   1742#  1742   1744#  1744   1749#  1749   1751#  1751   1753#  1753
         1755#  1755   1757#  1757   1759#  1759   1761#  1761   1763#  1763   1765#  1765   1767#  1767
         1773#  1773   1775#  1775   1777#  1777   1783#  1783   1785#  1785   1787#  1787   1789#  1789
         1791#  1791   1793#  1793   1795#  1795   1797#  1797   1802#  1802   1804#  1804   1809#  1809
         1811#  1811   1813#  1813   1818#  1818   1823#  1823   1825#  1825   1975#  1975   1977#  1977
         1979#  1979   1981#  1981   1984#  1984   1986#  1986   1988#  1988   1990#  1990   1992#  1992
         1994#  1994   1997#  1997   2002#  2002   2008#  2008   2010#  2010   2012#  2012   2014#  2014
         2016#  2016   2018#  2018   2020#  2020   2022#  2022   2027#  2027   2029#  2029   2031#  2031
         2033#  2033   2035#  2035   2037#  2037   2039#  2039   2041#  2041   2043#  2043   2045#  2045
         2047#  2047   2049#  2049   2051#  2051   2053#  2053   2055#  2055   2057#  2057   2059#  2059
         2061#  2061   2063#  2063   2065#  2065   2070#  2070   2072#  2072   2074#  2074   2080#  2080
         2082#  2082   2084#  2084   2086#  2086   2088#  2088   2090#  2090   2092#  2092   2094#  2094
         2096#  2096   2098#  2098   2100#  2100   2102#  2102   2107#  2107   2109#  2109   2269#  2269
         2272#  2272   2274#  2274   2276#  2276   2278#  2278   2280#  2280   2283#  2283   2285#  2285
         2287#  2287   2289#  2289   2292#  2292   2298#  2298   2300#  2300   2302#  2302   2308#  2308
         2310#  2310   2312#  2312   2314#  2314   2316#  2316   2318#  2318   2320#  2320   2322#  2322
         2327#  2327   2329#  2329   2331#  2331   2333#  2333   2335#  2335   2337#  2337   2339#  2339
         2341#  2341   2343#  2343   2345#  2345   2347#  2347   2349#  2349   2351#  2351   2353#  2353
         2355#  2355   2357#  2357   2359#  2359   2361#  2361   2363#  2363   2365#  2365   2370#  2370
         2372#  2372   2374#  2374   2380#  2380   2382#  2382   2384#  2384   2386#  2386   2388#  2388
         2390#  2390   2392#  2392   2394#  2394   2396#  2396   2398#  2398   2400#  2400   2402#  2402
         2407#  2407   2409#  2409   2485#  2485   2487#  2487   2594#  2594   2596#  2596   2598#  2598
         2600#  2600   2602#  2602   2855#  2855   2857#  2857   2859#  2859   2861#  2861   2866#  2866
         2868#  2868   2870#  2870   2872#  2872   2874#  2874   2876#  2876   2881#  2881   2883#  2883
         2885#  2885   2890#  2890   2892#  2892   2897#  2897   2899#  2899   2901#  2901   2903#  2903
         2905#  2905   2907#  2907   2909#  2909   2911#  2911   2913#  2913   3165#  3165   3167#  3167
         3169#  3169   3171#  3171   3176#  3176   3178#  3178   3180#  3180   3182#  3182   3184#  3184
         3186#  3186   3191#  3191   3193#  3193   3195#  3195   3200#  3200   3202#  3202   3207#  3207
         3209#  3209   3211#  3211   3213#  3213   3215#  3215   3217#  3217   3219#  3219   3221#  3221
         3223#  3223   3482#  3482   3484#  3484   3486#  3486   3488#  3488   3493#  3493   3495#  3495
         3497#  3497   3499#  3499   3501#  3501   3503#  3503   3508#  3508   3510#  3510   3512#  3512
         3514#  3514   3516#  3516   3518#  3518   3520#  3520   3525#  3525   3527#  3527   3532#  3532
         3534#  3534   3536#  3536   3538#  3538   3540#  3540   3542#  3542   3544#  3544   3546#  3546
         3548#  3548   3805#  3805   3807#  3807   3809#  3809   3811#  3811   3816#  3816   3818#  3818
         3820#  3820   3822#  3822   3824#  3824   3826#  3826   3831#  3831   3833#  3833   3835#  3835
         3837#  3837   3839#  3839   3841#  3841   3843#  3843   3848#  3848   3850#  3850   3855#  3855
         3857#  3857   3859#  3859   3861#  3861   3863#  3863   3865#  3865   3867#  3867   3869#  3869
         3871#  3871   4128#  4128   4130#  4130   4132#  4132   4134#  4134   4139#  4139   4141#  4141
         4143#  4143   4145#  4145   4147#  4147   4149#  4149   4154#  4154   4156#  4156   4158#  4158
         4160#  4160   4162#  4162   4164#  4164   4166#  4166   4171#  4171   4173#  4173   4178#  4178
         4180#  4180   4182#  4182   4184#  4184   4186#  4186   4188#  4188   4190#  4190   4192#  4192
         4194#  4194   4452#  4452   4454#  4454   4456#  4456   4458#  4458   4463#  4463   4465#  4465
         4467#  4467   4469#  4469   4471#  4471   4473#  4473   4478#  4478   4480#  4480   4482#  4482
         4484#  4484   4486#  4486   4488#  4488   4490#  4490   4495#  4495   4497#  4497   4502#  4502
         4504#  4504   4506#  4506   4508#  4508   4510#  4510   4512#  4512   4514#  4514   4516#  4516
         4518#  4518   4740#  4740   4742#  4742   4744#  4744   4746#  4746   4751#  4751   4753#  4753
         4755#  4755   4757#  4757   4759#  4759   4761#  4761   4766#  4766   4768#  4768   4770#  4770
         4772#  4772   4774#  4774   4776#  4776   4778#  4778   4783#  4783   4785#  4785   4790#  4790
         4792#  4792   4794#  4794   4796#  4796   4798#  4798   4800#  4800   4802#  4802   4804#  4804
         4806#  4806   5027#  5027   5029#  5029   5031#  5031   5033#  5033   5038#  5039   5040#  5040
         5042#  5042   5044#  5044   5046#  5046   5048#  5048   5053#  5053   5055#  5055   5057#  5057
         5059#  5059   5061#  5061   5063#  5063   5065#  5065   5070#  5070   5072#  5072   5077#  5077
```

EQ 1818

SEQ 1831

5079#	5079	5081#	5081	5083#	5083	5085#	5085	5087#	5087	5089#	5089	5091#	5091
5093#	5093	5313#	5313	5315#	5315	5317#	5317	5319#	5319	5324#	5324	5326#	5326
5328#	5328	5330#	5330	5332#	5332	5334#	5334	5339#	5339	5341#	5341	5343#	5343
5345#	5345	5347#	5347	5349#	5349	5351#	5351	5356#	5356	5358#	5358	5363#	5363
5365#	5365	5367#	5367	5369#	5369	5371#	5371	5373#	5373	5375#	5375	5377#	5377
5379#	5379	5602#	5602	5604#	5604	5606#	5606	5608#	5608	5613#	5613	5615#	5615
5617#	5617	5619#	5619	5621#	5621	5623#	5623	5628#	5628	5630#	5630	5632#	5632
5634#	5634	5636#	5636	5638#	5638	5640#	5640	5645#	5645	5647#	5647	5652#	5652
5654#	5654	5656#	5656	5658#	5658	5660#	5660	5662#	5662	5664#	5664	5666#	5666
5668#	5668	5774#	5774	5776#	5776	5778#	5778	5780#	5780	5782#	5782	6039#	6039
6041#	6041	6043#	6043	6045#	6045	6050#	6050	6052#	6052	6054#	6054	6056#	6056
6058#	6058	6060#	6060	6062#	6062	6064#	6064	6069#	6069	6071#	6071	6073#	6073
6078#	6078	6080#	6080	6085#	6085	6087#	6087	6089#	6089	6091#	6091	6093#	6093
6095#	6095	6097#	6097	6099#	6099	6101#	6101	6359#	6359	6361#	6361	6363#	6363
6365#	6365	6370#	6370	6372#	6372	6374#	6374	6376#	6376	6378#	6378	6380#	6380
6382#	6382	6384#	6384	6389#	6389	6391#	6391	6393#	6393	6395#	6395	6397#	6397
6399#	6399	6401#	6401	6406#	6406	6408#	6408	6413#	6413	6415#	6415	6417#	6417
6419#	6419	6421#	6421	6423#	6423	6425#	6425	6427#	6427	6429#	6429	6685#	6685
6687#	6687	6689#	6689	6691#	6691	6696#	6696	6698#	6698	6700#	6700	6702#	6702
6704#	6704	6706#	6706	6708#	6708	6710#	6710	6715#	6715	6717#	6717	6719#	6719
6721#	6721	6723#	6723	6725#	6725	6727#	6727	6732#	6732	6734#	6734	6739#	6739
6741#	6741	6743#	6743	6745#	6745	6747#	6747	6749#	6749	6751#	6751	6753#	6753
6755#	6755	7013#	7013	7015#	7015	7017#	7017	7019#	7019	7024#	7024	7026#	7026
7028#	7028	7030#	7030	7032#	7032	7034#	7034	7036#	7036	7038#	7038	7043#	7043
7045#	7045	7047#	7047	7049#	7049	7051#	7051	7053#	7053	7055#	7055	7060#	7060
7062#	7062	7067#	7067	7069#	7069	7071#	7071	7073#	7073	7075#	7075	7077#	7077
7079#	7079	7081#	7081	7083#	7083	7341#	7341	7343#	7343	7345#	7345	7347#	7347
7352#	7352	7354#	7354	7356#	7356	7358#	7358	7360#	7360	7362#	7362	7364#	7364
7366#	7366	7371#	7371	7373#	7373	7375#	7375	7377#	7377	7379#	7379	7381#	7381
7383#	7383	7388#	7388	7390#	7390	7395#	7395	7397#	7397	7399#	7399	7401#	7401
7403#	7403	7405#	7405	7407#	7407	7409#	7409	7411#	7411	7633#	7633	7635#	7635
7637#	7637	7639#	7639	7644#	7644	7646#	7646	7648#	7648	7650#	7650	7652#	7652
7654#	7654	7656#	7656	7658#	7658	7663#	7663	7665#	7665	7667#	7667	7669#	7669
7671#	7671	7673#	7673	7675#	7675	7680#	7680	7682#	7682	7687#	7687	7689#	7689
7691#	7691	7693#	7693	7695#	7695	7697#	7697	7699#	7699	7701#	7701	7703#	7703
7924#	7924	7926#	7926	7928#	7928	7930#	7930	7935#	7935	7937#	7937	7939#	7939
7941#	7941	7943#	7943	7945#	7945	7947#	7947	7949#	7949	7954#	7954	7956#	7956
7958#	7958	7960#	7960	7962#	7962	7964#	7964	7966#	7966	7971#	7971	7973#	7973
7978#	7978	7980#	7980	7982#	7982	7984#	7984	7986#	7986	7988#	7988	7990#	7990
7992#	7992	7994#	7994	8214#	8214	8216#	8216	8218#	8218	8220#	8220	8225#	8225
8227#	8227	8229#	8229	8231#	8231	8233#	8233	8235#	8235	8237#	8237	8239#	8239
8244#	8244	8246#	8246	8248#	8248	8250#	8250	8252#	8252	8254#	8254	8256#	8256
8261#	8261	8263#	8263	8268#	8268	8270#	8270	8272#	8272	8274#	8274	8276#	8276
8278#	8278	8280#	8280	8282#	8282	8284#	8284	8503#	8503	8505#	8505	8507#	8507
8509#	8509	8514#	8514	8516#	8516	8518#	8518	8520#	8520	8522#	8522	8524#	8524
8526#	8526	8528#	8528	8533#	8533	8535#	8535	8537#	8537	8539#	8539	8541#	8541
8543#	8543	8545#	8545	8550#	8550	8552#	8552	8557#	8557	8559#	8559	8561#	8561
8563#	8563	8565#	8565	8567#	8567	8569#	8569	8571#	8571	8573#	8573	8758#	8758
8760#	8760	8762#	8762	8764#	8764	8766#	8766	8768#	8768	8770#	8770	8772#	8772
8774#	8774	8776#	8776	8778#	8778	8780#	8780	8782#	8782	8784#	8784	8786#	8786
8791#	8791	8793#	8793	8795#	8795	8797#	8797	8799#	8799	8801#	8801	8803#	8803
8805#	8805	8810#	8810	8812#	8812	8814#	8814	8819#	8819	8821#	8821	8823#	8823
8825#	8825	8830#	8830	8832#	8832	8837#	8837	8839#	8839	8841#	8841	8843#	8843

```
EQ 1819      8845#  8845    8847#  8847    8849#  8849    8851#  8851    8853#  8853    8858#  8858    8860#  8860   SEQ 1832
             8862#  8862    8864#  8864    8866#  8866    8868#  8868    8870#  8870    8872#  8872    8874#  8874
$CHR          194#   194     196#   196     198#   198     200#   200     202#   202     204#   204     206#   206
              208#   208     210#   210     212#   212     214#   214     216#   216     218#   218     220#   220
              222#   222     224#   224     226#   226     228#   228     230#   230     232#   232     234#   234
              236#   236     238#   238     240#   240     242#   242     244#   244     246#   246     248#   248
              250#   250     252#   252     254#   254     256#   256     258#   258     260#   260     262#   262
              264#   264     266#   266     268#   268     273#   273     275#   275     280#   280     282#   282
              287#   287     289#   289     291#   291     296#   296     298#   298     300#   300     305#   305
              307#   307     309#   309     314#   314     316#   316     318#   318     323#   323     325#   325
              327#   327     332#   332     334#   334     336#   336     341#   341     343#   343     345#   345
              350#   350     352#   352     354#   354     359#   359     361#   361     363#   363     368#   368
              370#   370     372#   372     377#   377     379#   379     381#   381     386#   386     388#   388
              390#   390     395#   395     397#   397     399#   399     404#   404     406#   406     408#   408
              413#   413     415#   415     417#   417     422#   422     424#   424     426#   426     431#   431
              433#   433     435#   435     440#   440     442#   442     444#   444     449#   449     451#   451
              453#   453     458#   458     460#   460     462#   462     467#   467     469#   469     471#   471
              476#   476     478#   478     480#   480     485#   485     487#   487     489#   489     494#   494
              496#   496     498#   498     503#   503     505#   505     507#   507     512#   512     514#   514
              516#   516     521#   521     523#   523     525#   525     530#   530     532#   532     534#   534
              539#   539     541#   541     543#   543     548#   548     550#   550     552#   552     557#   557
              559#   559     561#   561     566#   566     568#   568     570#   570     575#   575     577#   577
              579#   579     584#   584     586#   586     588#   588     593#   593     595#   595     597#   597
              599#   599     604#   604     609#   609     611#   611     613#   613     615#   615     617#   617
              717#   717     719#   719     721#   721     723#   723     725#   725     727#   727     729#   729
              731#   731     733#   733     735#   735     740#   740     742#   742     747#   747     749#   749
              754#   754     756#   756     758#   758     763#   763     765#   765     767#   767     772#   772
              774#   774     776#   776     781#   781     783#   783     785#   785     790#   790     792#   792
              794#   794     799#   799     801#   801     803#   803     808#   808     810#   810     812#   812
              817#   817     822#   822     824#   824     826#   826     828#   828     830#   830     832#   832
              938#   938     943#   943     948#   948     950#   950     955#   955     957#   957     959#   959
              964#   964     966#   966     971#   971     973#   973    1062#  1062    1064#  1064    1066#  1066
             1071#  1071    1076#  1076    1081#  1081    1083#  1083    1088#  1088    1093#  1093    1095#  1095
             1100#  1100    1102#  1102    1108#  1108    1110#  1110    1112#  1112    1114#  1114    1116#  1116
             1118#  1118    1120#  1120    1122#  1122    1127#  1127    1129#  1129    1330#  1330    1332#  1332
             1334#  1334    1336#  1336    1341#  1341    1343#  1343    1345#  1345    1347#  1347    1349#  1349
             1351#  1351    1356#  1356    1358#  1358    1360#  1360    1365#  1365    1371#  1371    1373#  1373
             1375#  1375    1377#  1377    1379#  1379    1381#  1381    1383#  1383    1385#  1385    1392#  1392
             1394#  1394    1396#  1396    1398#  1398    1400#  1400    1402#  1402    1404#  1404    1406#  1406
             1408#  1408    1410#  1410    1412#  1412    1414#  1414    1416#  1416    1421#  1421    1423#  1423
             1428#  1428    1430#  1430    1432#  1432    1437#  1437    1439#  1439    1444#  1444    1446#  1446
             1451#  1451    1456#  1456    1458#  1458    1586#  1586    1588#  1588    1590#  1590    1592#  1592
             1676#  1676    1678#  1678    1680#  1680    1682#  1682    1684#  1684    1686#  1686    1691#  1691
             1693#  1693    1695#  1695    1697#  1697    1699#  1699    1701#  1701    1706#  1706    1708#  1708
             1710#  1710    1712#  1712    1714#  1714    1716#  1716    1718#  1718    1720#  1720    1722#  1722
             1724#  1724    1726#  1726    1728#  1728    1730#  1730    1732#  1732    1734#  1734    1736#  1736
             1738#  1738    1740#  1740    1742#  1742    1744#  1744    1749#  1749    1751#  1751    1753#  1753
             1755#  1755    1757#  1757    1759#  1759    1761#  1761    1763#  1763    1765#  1765    1767#  1767
             1773#  1773    1775#  1775    1777#  1777    1783#  1783    1785#  1785    1787#  1787    1789#  1789
             1791#  1791    1793#  1793    1795#  1795    1797#  1797    1802#  1802    1804#  1804    1809#  1809
             1811#  1811    1813#  1813    1818#  1818    1823#  1823    1825#  1825    1975#  1975    1977#  1977
             1979#  1979    1981#  1981    1984#  1984    1986#  1986    1988#  1988    1990#  1990    1992#  1992
             1994#  1994    1997#  1997    2002#  2002    2008#  2008    2010#  2010    2012#  2012    2014#  2014
```

```
2016#  2016   2018#  2018   2020#  2020   2022#  2022   2027#  2027   2029#  2029   2031#  2031
2033#  2033   2035#  2035   2037#  2037   2039#  2039   2041#  2041   2043#  2043   2045#  2045
2047#  2047   2049#  2049   2051#  2051   2053#  2053   2055#  2055   2057#  2057   2059#  2059
2061#  2061   2063#  2063   2065#  2065   2070#  2070   2072#  2072   2074#  2074   2080#  2080
2082#  2082   2084#  2084   2086#  2086   2088#  2088   2090#  2090   2092#  2092   2094#  2094
2096#  2096   2098#  2098   2100#  2100   2102#  2102   2107#  2107   2109#  2109   2269#  2269
2272#  2272   2274#  2274   2276#  2276   2278#  2278   2280#  2280   2283#  2283   2285#  2285
2287#  2287   2289#  2289   2292#  2292   2298#  2298   2300#  2300   2302#  2302   2308#  2308
2310#  2310   2312#  2312   2314#  2314   2316#  2316   2318#  2318   2320#  2320   2322#  2322
2327#  2327   2329#  2329   2331#  2331   2333#  2333   2335#  2335   2337#  2337   2339#  2339
2341#  2341   2343#  2343   2345#  2345   2347#  2347   2349#  2349   2351#  2351   2353#  2353
2355#  2355   2357#  2357   2359#  2359   2361#  2361   2363#  2363   2365#  2365   2370#  2370
2372#  2372   2374#  2374   2380#  2380   2382#  2382   2384#  2384   2386#  2386   2388#  2388
2390#  2390   2392#  2392   2394#  2394   2396#  2396   2398#  2398   2400#  2400   2402#  2402
2407#  2407   2409#  2409   2485#  2485   2487#  2487   2594#  2594   2596#  2596   2598#  2598
2600#  2600   2602#  2602   2855#  2855   2857#  2857   2859#  2859   2861#  2861   2866#  2866
2868#  2868   2870#  2870   2872#  2872   2874#  2874   2876#  2876   2881#  2881   2883#  2883
2885#  2885   2890#  2890   2892#  2892   2897#  2897   2899#  2899   2901#  2901   2903#  2903
2905#  2905   2907#  2907   2909#  2909   2911#  2911   2913#  2913   3165#  3165   3167#  3167
3169#  3169   3171#  3171   3176#  3176   3178#  3178   3180#  3180   3182#  3182   3184#  3184
3186#  3186   3191#  3191   3193#  3193   3195#  3195   3200#  3200   3202#  3202   3207#  3207
3209#  3209   3211#  3211   3213#  3213   3215#  3215   3217#  3217   3219#  3219   3221#  3221
3223#  3223   3482#  3482   3484#  3484   3486#  3486   3488#  3488   3493#  3493   3495#  3495
3497#  3497   3499#  3499   3501#  3501   3503#  3503   3508#  3508   3510#  3510   3512#  3512
3514#  3514   3516#  3516   3518#  3518   3520#  3520   3525#  3525   3527#  3527   3532#  3532
3534#  3534   3536#  3536   3538#  3538   3540#  3540   3542#  3542   3544#  3544   3546#  3546
3548#  3548   3805#  3805   3807#  3807   3809#  3809   3811#  3811   3816#  3816   3818#  3818
3820#  3820   3822#  3822   3824#  3824   3826#  3826   3831#  3831   3833#  3833   3835#  3835
3837#  3837   3839#  3839   3841#  3841   3843#  3843   3848#  3848   3850#  3850   3855#  3855
3857#  3857   3859#  3859   3861#  3861   3863#  3863   3865#  3865   3867#  3867   3869#  3869
3871#  3871   4128#  4128   4130#  4130   4132#  4132   4134#  4134   4139#  4139   4141#  4141
4143#  4143   4145#  4145   4147#  4147   4149#  4149   4154#  4154   4156#  4156   4158#  4158
4160#  4160   4162#  4162   4164#  4164   4166#  4166   4171#  4171   4173#  4173   4178#  4178
4180#  4180   4182#  4182   4184#  4184   4186#  4186   4188#  4188   4190#  4190   4192#  4192
4194#  4194   4452#  4452   4454#  4454   4456#  4456   4458#  4458   4463#  4463   4465#  4465
4467#  4467   4469#  4469   4471#  4471   4473#  4473   4478#  4478   4480#  4480   4482#  4482
4484#  4484   4486#  4486   4488#  4488   4490#  4490   4495#  4495   4497#  4497   4502#  4502
4504#  4504   4506#  4506   4508#  4508   4510#  4510   4512#  4512   4514#  4514   4516#  4516
4518#  4518   4740#  4740   4742#  4742   4744#  4744   4746#  4746   4751#  4751   4753#  4753
4755#  4755   4757#  4757   4759#  4759   4761#  4761   4766#  4766   4768#  4768   4770#  4770
4772#  4772   4774#  4774   4776#  4776   4778#  4778   4783#  4783   4785#  4785   4790#  4790
4792#  4792   4794#  4794   4796#  4796   4798#  4798   4800#  4800   4802#  4802   4804#  4804
4806#  4806   5027#  5027   5029#  5029   5031#  5031   5033#  5033   5038#  5038   5040#  5040
5042#  5042   5044#  5044   5046#  5046   5048#  5048   5053#  5053   5055#  5055   5057#  5057
5059#  5059   5061#  5061   5063#  5063   5065#  5065   5070#  5070   5072#  5072   5077#  5077
5079#  5079   5081#  5081   5083#  5083   5085#  5085   5087#  5087   5089#  5089   5091#  5091
5093#  5093   5313#  5313   5315#  5315   5317#  5317   5319#  5319   5324#  5324   5326#  5326
5328#  5328   5330#  5330   5332#  5332   5334#  5334   5339#  5339   5341#  5341   5343#  5343
5345#  5345   5347#  5347   5349#  5349   5351#  5351   5356#  5356   5358#  5358   5363#  5363
5365#  5365   5367#  5367   5369#  5369   5371#  5371   5373#  5373   5375#  5375   5377#  5377
5379#  5379   5602#  5602   5604#  5604   5606#  5606   5608#  5608   5613#  5613   5615#  5615
5617#  5617   5619#  5619   5621#  5621   5623#  5623   5628#  5628   5630#  5630   5632#  5632
5634#  5634   5636#  5636   5638#  5638   5640#  5640   5645#  5645   5647#  5647   5652#  5652
```

SEQ 1821

```
           5654#  5654   5656#  5656   5658#  5658   5660#  5660   5662#  5662   5664#  5664   5666#  5666
           5668#  5668   5774#  5774   5776#  5776   5778#  5778   5780#  5780   5782#  5782   6039#  6039
           6041#  6041   6043#  6043   6045#  6045   6050#  6050   6052#  6052   6054#  6054   6056#  6056
           6058#  6058   6060#  6060   6062#  6062   6064#  6064   6069#  6069   6071#  6071   6073#  6073
           6078#  6078   6080#  6080   6085#  6085   6087#  6087   6089#  6089   6091#  6091   6093#  6093
           6095#  6095   6097#  6097   6099#  6099   6101#  6101   6359#  6359   6361#  6361   6363#  6363
           6365#  6365   6370#  6370   6372#  6372   6374#  6374   6376#  6376   6378#  6378   6380#  6380
           6382#  6382   6384#  6384   6389#  6389   6391#  6391   6393#  6393   6395#  6395   6397#  6397
           6399#  6399   6401#  6401   6406#  6406   6408#  6408   6413#  6413   6415#  6415   6417#  6417
           6419#  6419   6421#  6421   6423#  6423   6425#  6425   6427#  6427   6429#  6429   6685#  6685
           6687#  6687   6689#  6689   6691#  6691   6696#  6696   6698#  6698   6700#  6700   6702#  6702
           6704#  6704   6706#  6706   6708#  6708   6710#  6710   6715#  6715   6717#  6717   6719#  6719
           6721#  6721   6723#  6723   6725#  6725   6727#  6727   6732#  6732   6734#  6734   6739#  6739
           6741#  6741   6743#  6743   6745#  6745   6747#  6747   6749#  6749   6751#  6751   6753#  6753
           6755#  6755   7013#  7013   7015#  7015   7017#  7017   7019#  7019   7024#  7024   7026#  7026
           7028#  7028   7030#  7030   7032#  7032   7034#  7034   7036#  7036   7038#  7038   7043#  7043
           7045#  7045   7047#  7047   7049#  7049   7051#  7051   7053#  7053   7055#  7055   7060#  7060
           7062#  7062   7067#  7067   7069#  7069   7071#  7071   7073#  7073   7075#  7075   7077#  7077
           7079#  7079   7081#  7081   7083#  7083   7341#  7341   7343#  7343   7345#  7345   7347#  7347
           7352#  7352   7354#  7354   7356#  7356   7358#  7358   7360#  7360   7362#  7362   7364#  7364
           7366#  7366   7371#  7371   7373#  7373   7375#  7375   7377#  7377   7379#  7379   7381#  7381
           7383#  7383   7388#  7388   7390#  7390   7395#  7395   7397#  7397   7399#  7399   7401#  7401
           7403#  7403   7405#  7405   7407#  7407   7409#  7409   7411#  7411   7633#  7633   7635#  7635
           7637#  7637   7639#  7639   7644#  7644   7646#  7646   7648#  7648   7650#  7650   7652#  7652
           7654#  7654   7656#  7656   7658#  7658   7663#  7663   7665#  7665   7667#  7667   7669#  7669
           7671#  7671   7673#  7673   7675#  7675   7680#  7680   7682#  7682   7687#  7687   7689#  7689
           7691#  7691   7693#  7693   7695#  7695   7697#  7697   7699#  7699   7701#  7701   7703#  7703
           7924#  7924   7926#  7926   7928#  7928   7930#  7930   7935#  7935   7937#  7937   7939#  7939
           7941#  7941   7943#  7943   7945#  7945   7947#  7947   7949#  7949   7954#  7954   7956#  7956
           7958#  7958   7960#  7960   7962#  7962   7964#  7964   7966#  7966   7971#  7971   7973#  7973
           7978#  7978   7980#  7980   7982#  7982   7984#  7984   7986#  7986   7988#  7988   7990#  7990
           7992#  7992   7994#  7994   8214#  8214   8216#  8216   8218#  8218   8220#  8220   8225#  8225
           8227#  8227   8229#  8229   8231#  8231   8233#  8233   8235#  8235   8237#  8237   8239#  8239
           8244#  8244   8246#  8246   8248#  8248   8250#  8250   8252#  8252   8254#  8254   8256#  8256
           8261#  8261   8263#  8263   8268#  8268   8270#  8270   8272#  8272   8274#  8274   8276#  8276
           8278#  8278   8280#  8280   8282#  8282   8284#  8284   8503#  8503   8505#  8505   8507#  8507
           8509#  8509   8514#  8514   8516#  8516   8518#  8518   8520#  8520   8522#  8522   8524#  8524
           8526#  8526   8528#  8528   8533#  8533   8535#  8535   8537#  8537   8539#  8539   8541#  8541
           8543#  8543   8545#  8545   8550#  8550   8552#  8552   8557#  8557   8559#  8559   8561#  8561
           8563#  8563   8565#  8565   8567#  8567   8569#  8569   8571#  8571   8573#  8573   8758#  8758
           8760#  8760   8762#  8762   8764#  8764   8766#  8766   8768#  8768   8770#  8770   8772#  8772
           8774#  8774   8776#  8776   8778#  8778   8780#  8780   8782#  8782   8784#  8784   8786#  8786
           8791#  8791   8793#  8793   8795#  8795   8797#  8797   8799#  8799   8801#  8801   8803#  8803
           8805#  8805   8810#  8810   8812#  8812   8814#  8814   8819#  8819   8821#  8821   8823#  8823
           8825#  8825   8830#  8830   8832#  8832   8837#  8837   8839#  8839   8841#  8841   8843#  8843
           8845#  8845   8847#  8847   8849#  8849   8851#  8851   8853#  8853   8858#  8858   8860#  8860
           8862#  8862   8864#  8864   8866#  8866   8868#  8868   8870#  8870   8872#  8872   8874#  8874
$GARG       194#   194    196#   196    198#   198    200#   200    202#   202    204#   204    206#   206
            208#   208    210#   210    212#   212    214#   214    216#   216    218#   218    220#   220
            222#   222    224#   224    226#   226    228#   228    230#   230    232#   232    234#   234
            236#   236    238#   238    240#   240    242#   242    244#   244    246#   246    248#   248
            250#   250    252#   252    254#   254    256#   256    258#   258    260#   260    262#   262
            264#   264    266#   266    268#   268    273#   273    275#   275    280#   280    282#   282
```

SEQ 1834

```
 287#  287   289#  289   291#  291   296#  296   298#  298   300#  300   305#  305   SEQ 1835
 307#  307   309#  309   314#  314   316#  316   318#  318   323#  323   325#  325
 327#  327   332#  332   334#  334   336#  336   341#  341   343#  343   345#  345
 350#  350   352#  352   354#  354   359#  359   361#  361   363#  363   368#  368
 370#  370   372#  372   377#  377   379#  379   381#  381   386#  386   388#  388
 390#  390   395#  395   397#  397   399#  399   404#  404   406#  406   408#  408
 413#  413   415#  415   417#  417   422#  422   424#  424   426#  426   431#  431
 433#  433   435#  435   440#  440   442#  442   444#  444   449#  449   451#  451
 453#  453   458#  458   460#  460   462#  462   467#  467   469#  469   471#  471
 476#  476   478#  478   480#  480   485#  485   487#  487   489#  489   494#  494
 496#  496   498#  498   503#  503   505#  505   507#  507   512#  512   514#  514
 516#  516   521#  521   523#  523   525#  525   530#  530   532#  532   534#  534
 539#  539   541#  541   543#  543   548#  548   550#  550   552#  552   557#  557
 559#  559   561#  561   566#  566   568#  568   570#  570   575#  575   577#  577
 579#  579   584#  584   586#  586   588#  588   593#  593   595#  595   597#  597
 599#  599   604#  604   609#  609   611#  611   613#  613   615#  615   617#  617
 717#  717   719#  719   721#  721   723#  723   725#  725   727#  727   729#  729
 731#  731   733#  733   735#  735   740#  740   742#  742   747#  747   749#  749
 754#  754   756#  756   758#  758   763#  763   765#  765   767#  767   772#  772
 774#  774   776#  776   781#  781   783#  783   785#  785   790#  790   792#  792
 794#  794   799#  799   801#  801   803#  803   808#  808   810#  810   812#  812
 817#  817   822#  822   824#  824   826#  826   828#  828   830#  830   832#  832
 938#  938   943#  943   948#  948   950#  950   955#  955   957#  957   959#  959
 964#  964   966#  966   971#  971   973#  973  1062# 1062  1064# 1064  1066# 1066
1071# 1071  1076# 1076  1081# 1081  1083# 1083  1088# 1088  1093# 1093  1095# 1095
1100# 1100  1102# 1102  1108# 1108  1110# 1110  1112# 1112  1114# 1114  1116# 1116
1118# 1118  1120# 1120  1122# 1122  1127# 1127  1129# 1129  1330# 1330  1332# 1332
1334# 1334  1336# 1336  1341# 1341  1343# 1343  1345# 1345  1347# 1347  1349# 1349
1351# 1351  1356# 1356  1358# 1358  1360# 1360  1365# 1365  1371# 1371  1373# 1373
1375# 1375  1377# 1377  1379# 1379  1381# 1381  1383# 1383  1385# 1385  1392# 1392
1394# 1394  1396# 1396  1398# 1398  1400# 1400  1402# 1402  1404# 1404  1406# 1406
1408# 1408  1410# 1410  1412# 1412  1414# 1414  1416# 1416  1421# 1421  1423# 1423
1428# 1428  1430# 1430  1432# 1432  1437# 1437  1439# 1439  1444# 1444  1446# 1446
1451# 1451  1456# 1456  1458# 1458  1586# 1586  1588# 1588  1590# 1590  1592# 1592
1676# 1676  1678# 1678  1680# 1680  1682# 1682  1684# 1684  1686# 1686  1691# 1691
1693# 1693  1695# 1695  1697# 1697  1699# 1699  1701# 1701  1706# 1706  1708# 1708
1710# 1710  1712# 1712  1714# 1714  1716# 1716  1718# 1718  1720# 1720  1722# 1722
1724# 1724  1726# 1726  1728# 1728  1730# 1730  1732# 1732  1734# 1734  1736# 1736
1738# 1738  1740# 1740  1742# 1742  1744# 1744  1749# 1749  1751# 1751  1753# 1753
1755# 1755  1757# 1757  1759# 1759  1761# 1761  1763# 1763  1765# 1765  1767# 1767
1773# 1773  1775# 1775  1777# 1777  1783# 1783  1785# 1785  1787# 1787  1789# 1789
1791# 1791  1793# 1793  1795# 1795  1797# 1797  1802# 1802  1804# 1804  1809# 1809
1811# 1811  1813# 1813  1818# 1818  1823# 1823  1825# 1825  1975# 1975  1977# 1977
1979# 1979  1981# 1981  1984# 1984  1986# 1986  1988# 1988  1990# 1990  1992# 1992
1994# 1994  1997# 1997  2002# 2002  2008# 2008  2010# 2010  2012# 2012  2014# 2014
2016# 2016  2018# 2018  2020# 2020  2022# 2022  2027# 2027  2029# 2029  2031# 2031
2033# 2033  2035# 2035  2037# 2037  2039# 2039  2041# 2041  2043# 2043  2045# 2045
2047# 2047  2049# 2049  2051# 2051  2053# 2053  2055# 2055  2057# 2057  2059# 2059
2061# 2061  2063# 2063  2065# 2065  2070# 2070  2072# 2072  2074# 2074  2080# 2080
2082# 2082  2084# 2084  2086# 2086  2088# 2088  2090# 2090  2092# 2092  2094# 2094
2096# 2096  2098# 2098  2100# 2100  2102# 2102  2107# 2107  2109# 2109  2269# 2269
2272# 2272  2274# 2274  2276# 2276  2278# 2278  2280# 2280  2283# 2283  2285# 2285
2287# 2287  2289# 2289  2292# 2292  2298# 2298  2300# 2300  2302# 2302  2308# 2308
```

EQ 1823

```
2310#  2310   2312#  2312   2314#  2314   2316#  2316   2318#  2318   2320#  2320   2322#  2322
2327#  2327   2329#  2329   2331#  2331   2333#  2333   2335#  2335   2337#  2337   2339#  2339
2341#  2341   2343#  2343   2345#  2345   2347#  2347   2349#  2349   2351#  2351   2353#  2353
2355#  2355   2357#  2357   2359#  2359   2361#  2361   2363#  2363   2365#  2365   2370#  2370
2372#  2372   2374#  2374   2380#  2380   2382#  2382   2384#  2384   2386#  2386   2388#  2388
2390#  2390   2392#  2392   2394#  2394   2396#  2396   2398#  2398   2400#  2400   2402#  2402
2407#  2407   2409#  2409   2485#  2485   2487#  2487   2594#  2594   2596#  2596   2598#  2598
2600#  2600   2602#  2602   2855#  2855   2857#  2857   2859#  2859   2861#  2861   2866#  2866
2868#  2868   2870#  2870   2872#  2872   2874#  2874   2876#  2876   2881#  2881   2883#  2883
2885#  2885   2890#  2890   2892#  2892   2897#  2897   2899#  2899   2901#  2901   2903#  2903
2905#  2905   2907#  2907   2909#  2909   2911#  2911   2913#  2913   3165#  3165   3167#  3167
3169#  3169   3171#  3171   3176#  3176   3178#  3178   3180#  3180   3182#  3182   3184#  3184
3186#  3186   3191#  3191   3193#  3193   3195#  3195   3200#  3200   3202#  3202   3207#  3207
3209#  3209   3211#  3211   3213#  3213   3215#  3215   3217#  3217   3219#  3219   3221#  3221
3223#  3223   3482#  3482   3484#  3484   3486#  3486   3488#  3488   3493#  3493   3495#  3495
3497#  3497   3499#  3499   3501#  3501   3503#  3503   3508#  3508   3510#  3510   3512#  3512
3514#  3514   3516#  3516   3518#  3518   3520#  3520   3525#  3525   3527#  3527   3532#  3532
3534#  3534   3536#  3536   3538#  3538   3540#  3540   3542#  3542   3544#  3544   3546#  3546
3548#  3548   3805#  3805   3807#  3807   3809#  3809   3811#  3811   3816#  3816   3818#  3818
3820#  3820   3822#  3822   3824#  3824   3826#  3826   3831#  3831   3833#  3833   3835#  3835
3837#  3837   3839#  3839   3841#  3841   3843#  3843   3848#  3848   3850#  3850   3855#  3855
3857#  3857   3859#  3859   3861#  3861   3863#  3863   3865#  3865   3867#  3867   3869#  3869
3871#  3871   4128#  4128   4130#  4130   4132#  4132   4134#  4134   4139#  4139   4141#  4141
4143#  4143   4145#  4145   4147#  4147   4149#  4149   4154#  4154   4156#  4156   4158#  4158
4160#  4160   4162#  4162   4164#  4164   4166#  4166   4171#  4171   4173#  4173   4178#  4178
4180#  4180   4182#  4182   4184#  4184   4186#  4186   4188#  4188   4190#  4190   4192#  4192
4194#  4194   4452#  4452   4454#  4454   4456#  4456   4458#  4458   4463#  4463   4465#  4465
4467#  4467   4469#  4469   4471#  4471   4473#  4473   4478#  4478   4480#  4480   4482#  4482
4484#  4484   4486#  4486   4488#  4488   4490#  4490   4495#  4495   4497#  4497   4502#  4502
4504#  4504   4506#  4506   4508#  4508   4510#  4510   4512#  4512   4514#  4514   4516#  4516
4518#  4518   4740#  4740   4742#  4742   4744#  4744   4746#  4746   4751#  4751   4753#  4753
4755#  4755   4757#  4757   4759#  4759   4761#  4761   4766#  4766   4768#  4768   4770#  4770
4772#  4772   4774#  4774   4776#  4776   4778#  4778   4783#  4783   4785#  4785   4790#  4790
4792#  4792   4794#  4794   4796#  4796   4798#  4798   4800#  4800   4802#  4802   4804#  4804
4806#  4806   5027#  5027   5029#  5029   5031#  5031   5033#  5033   5038#  5038   5040#  5040
5042#  5042   5044#  5044   5046#  5046   5048#  5048   5053#  5053   5055#  5055   5057#  5057
5059#  5059   5061#  5061   5063#  5063   5065#  5065   5070#  5070   5072#  5072   5077#  5077
5079#  5079   5081#  5081   5083#  5083   5085#  5085   5087#  5087   5089#  5089   5091#  5091
5093#  5093   5313#  5313   5315#  5315   5317#  5317   5319#  5319   5324#  5324   5326#  5326
5328#  5328   5330#  5330   5332#  5332   5334#  5334   5339#  5339   5341#  5341   5343#  5343
5345#  5345   5347#  5347   5349#  5349   5351#  5351   5356#  5356   5358#  5358   5363#  5363
5365#  5365   5367#  5367   5369#  5369   5371#  5371   5373#  5373   5375#  5375   5377#  5377
5379#  5379   5602#  5602   5604#  5604   5606#  5606   5608#  5608   5613#  5613   5615#  5615
5617#  5617   5619#  5619   5621#  5621   5623#  5623   5628#  5628   5630#  5630   5632#  5632
5634#  5634   5636#  5636   5638#  5638   5640#  5640   5645#  5645   5647#  5647   5652#  5652
5654#  5654   5656#  5656   5658#  5658   5660#  5660   5662#  5662   5664#  5664   5666#  5666
5668#  5668   5774#  5774   5776#  5776   5778#  5778   5780#  5780   5782#  5782   6039#  6039
6041#  6041   6043#  6043   6045#  6045   6050#  6050   6052#  6052   6054#  6054   6056#  6056
6058#  6058   6060#  6060   6062#  6062   6064#  6064   6069#  6069   6071#  6071   6073#  6073
6078#  6078   6080#  6080   6085#  6085   6087#  6087   6089#  6089   6091#  6091   6093#  6093
6095#  6095   6097#  6097   6099#  6099   6101#  6101   6359#  6359   6361#  6361   6363#  6363
6365#  6365   6370#  6370   6372#  6372   6374#  6374   6376#  6376   6378#  6378   6380#  6380
6382#  6382   6384#  6384   6389#  6389   6391#  6391   6393#  6393   6395#  6395   6397#  6397
```

SEQ 1836

```
EQ 1824  6399#    6399     6401#    6401     6406#    6406     6408#    6408     6413#    6413     6415#    6415     6417#    6417      SEQ 1837
         6419#    6419     6421#    6421     6423#    6423     6425#    6425     6427#    6427     6429#    6429     6685#    6685
         6687#    6687     6689#    6689     6691#    6691     6696#    6696     6698#    6698     6700#    6700     6702#    6702
         6704#    6704     6706#    6706     6708#    6708     6710#    6710     6715#    6715     6717#    6717     6719#    6719
         6721#    6721     6723#    6723     6725#    6725     6727#    6727     6732#    6732     6734#    6734     6739#    6739
         6741#    6741     6743#    6743     6745#    6745     6747#    6747     6749#    6749     6751#    6751     6753#    6753
         6755#    6755     7013#    7013     7015#    7015     7017#    7017     7019#    7019     7024#    7024     7026#    7026
         7028#    7028     7030#    7030     7032#    7032     7034#    7034     7036#    7036     7038#    7038     7043#    7043
         7045#    7045     7047#    7047     7049#    7049     7051#    7051     7053#    7053     7055#    7055     7060#    7060
         7062#    7062     7067#    7067     7069#    7069     7071#    7071     7073#    7073     7075#    7075     7077#    7077
         7079#    7079     7081#    7081     7083#    7083     7341#    7341     7343#    7343     7345#    7345     7347#    7347
         7352#    7352     7354#    7354     7356#    7356     7358#    7358     7360#    7360     7362#    7362     7364#    7364
         7366#    7366     7371#    7371     7373#    7373     7375#    7375     7377#    7377     7379#    7379     7381#    7381
         7383#    7383     7388#    7388     7390#    7390     7395#    7395     7397#    7397     7399#    7399     7401#    7401
         7403#    7403     7405#    7405     7407#    7407     7409#    7409     7411#    7411     7633#    7633     7635#    7635
         7637#    7637     7639#    7639     7644#    7644     7646#    7646     7648#    7648     7650#    7650     7652#    7652
         7654#    7654     7656#    7656     7658#    7658     7663#    7663     7665#    7665     7667#    7667     7669#    7669
         7671#    7671     7673#    7673     7675#    7675     7680#    7680     7682#    7682     7687#    7687     7689#    7689
         7691#    7691     7693#    7693     7695#    7695     7697#    7697     7699#    7699     7701#    7701     7703#    7703
         7924#    7924     7926#    7926     7928#    7928     7930#    7930     7935#    7935     7937#    7937     7939#    7939
         7941#    7941     7943#    7943     7945#    7945     7947#    7947     7949#    7949     7954#    7954     7956#    7956
         7958#    7958     7960#    7960     7962#    7962     7964#    7964     7966#    7966     7971#    7971     7973#    7973
         7978#    7978     7980#    7980     7982#    7982     7984#    7984     7986#    7986     7988#    7988     7990#    7990
         7992#    7992     7994#    7994     8214#    8214     8216#    8216     8218#    8218     8220#    8220     8225#    8225
         8227#    8227     8229#    8229     8231#    8231     8233#    8233     8235#    8235     8237#    8237     8239#    8239
         8244#    8244     8246#    8246     8248#    8248     8250#    8250     8252#    8252     8254#    8254     8256#    8256
         8261#    8261     8263#    8263     8268#    8268     8270#    8270     8272#    8272     8274#    8274     8276#    8276
         8278#    8278     8280#    8280     8282#    8282     8284#    8284     8503#    8503     8505#    8505     8507#    8507
         8509#    8509     8514#    8514     8516#    8516     8518#    8518     8520#    8520     8522#    8522     8524#    8524
         8526#    8526     8528#    8528     8533#    8533     8535#    8535     8537#    8537     8539#    8539     8541#    8541
         8543#    8543     8545#    8545     8550#    8550     8552#    8552     8557#    8557     8559#    8559     8561#    8561
         8563#    8563     8565#    8565     8567#    8567     8569#    8569     8571#    8571     8573#    8573     8758#    8758
         8760#    8760     8762#    8762     8764#    8764     8766#    8766     8768#    8768     8770#    8770     8772#    8772
         8774#    8774     8776#    8776     8778#    8778     8780#    8780     8782#    8782     8784#    8784     8786#    8786
         8791#    8791     8793#    8793     8795#    8795     8797#    8797     8799#    8799     8801#    8801     8803#    8803
         8805#    8805     8810#    8810     8812#    8812     8814#    8814     8819#    8819     8821#    8821     8823#    8823
         8825#    8825     8830#    8830     8832#    8832     8837#    8837     8839#    8839     8841#    8841     8843#    8843
         8845#    8845     8847#    8847     8849#    8849     8851#    8851     8853#    8853     8858#    8858     8860#    8860
         8862#    8862     8864#    8864     8866#    8866     8868#    8868     8870#    8870     8872#    8872     8874#    8874
%ADDR     194#     194      196      196#     198      198#     200      200#     202      202#     204      204#     206      206#
          208      208#     210      210#     212      212#     214      214#     216      216#     218      218#     220      220#
          222      222#     224      224#     226      226#     228      228#     230      230#     232      232#     234      234#
          236      236#     238      238#     240      240#     242      242#     244      244#     246      246#     248      248#
          250      250#     252      252#     254      254#     256      256#     258      258#     260      260#     262      262#
          264      264#     266      266#     268      268#     273#     273      275      275#     280#     280      282      282#
          287#     287      289      289#     291      291#     296#     296      298      298#     300      300#     305#     305
          307      307#     309      309#     314#     314      316      316#     318      318#     323#     323      325      325#
          327      327#     332#     332      334      334#     336      336#     341#     341      343      343#     345      345#
          350#     350      352      352#     354      354#     359#     359      361      361#     363      363#     368#     368
          370      370#     372      372#     377#     377      379      379#     381      381#     386#     386      388      388#
          390      390#     395#     395      397      397#     399      399#     404#     404      406      406#     408      408#
          413#     413      415      415#     417      417#     422#     422      424      424#     426      426#     431#     431
          433      433#     435      435#     440#     440      442      442#     444      444#     449#     449      451      451#
```

EQ 1825

SEQ 1838

```
453     453#    458#    458     460     460#    462     462#    467#    467     469     469#    471     471#
476#    476     478     478#    480     480#    485#    485     487     487#    489     489#    494#    494
496     496#    498     498#    503#    503     505     505#    507     507#    512#    512     514     514#
516     516#    521#    521     523     523#    525     525#    530#    530     532     532#    534     534#
539#    539     541     541#    543     543#    548#    548     550     550#    552     552#    557#    557
559     559#    561     561#    566#    566     568     568#    570     570#    575#    575     577     577#
579     579#    584#    584     586     586#    588     588#    593#    593     595     595#    597     597#
599     599#    604#    604     609#    609     611     611#    613     613#    615     615#    617     617#
717#    717     719     719#    721     721#    723     723#    725     725#    727     727#    729     729#
731     731#    733     733#    735     735#    740#    740     742     742#    747#    747     749     749#
754#    754     756     756#    758     758#    763#    763     765     765#    767     767#    772#    772
774     774#    776     776#    781#    781     783     783#    785     785#    790#    790     792     792#
794     794#    799#    799     801     801#    803     803#    808#    808     810     810#    812     812#
817#    817     822#    822     824     824#    826     826#    828     828#    830     830#    832     832#
938#    938     943     943#    948     948#    950     950#    955     955#    957     957#    959     959#
964     964#    966     966#    971#    971     973     973#   1062#   1062    1064    1064#   1066    1066#
1071    1071#   1076    1076#   1081    1081#   1083    1083#   1088    1088#   1093    1093#   1095    1095#
1100    1100#   1102    1102#   1108    1108#   1110    1110#   1112    1112#   1114    1114#   1116    1116#
1118    1118#   1120    1120#   1122    1122#   1127#   1127    1129    1129#   1330#   1330    1332    1332#
1334    1334#   1336    1336#   1341    1341#   1343    1343#   1345    1345#   1347    1347#   1349    1349#
1351    1351#   1356    1356#   1358    1358#   1360    1360#   1365    1365#   1371    1371#   1373    1373#
1375    1375#   1377    1377#   1379    1379#   1381    1381#   1383    1383#   1385    1385#   1392#   1392
1394    1394#   1396    1396#   1398    1398#   1400    1400#   1402    1402#   1404    1404#   1406    1406#
1408    1408#   1410    1410#   1412    1412#   1414    1414#   1416    1416#   1421    1421#   1423    1423#
1428    1428#   1430    1430#   1432    1432#   1437#   1437    1439    1439#   1444#   1444    1446    1446#
1451#   1451    1456#   1456    1458    1458#   1586#   1586    1588    1588#   1590    1590#   1592    1592#
1676#   1676    1678    1678#   1680    1680#   1682    1682#   1684    1684#   1686    1686#   1691#   1691
1693    1693#   1695    1695#   1697    1697#   1699    1699#   1701    1701#   1706#   1706    1708    1708#
1710    1710#   1712    1712#   1714    1714#   1716    1716#   1718    1718#   1720    1720#   1722    1722#
1724    1724#   1726    1726#   1728    1728#   1730    1730#   1732    1732#   1734    1734#   1736    1736#
1738    1738#   1740    1740#   1742    1742#   1744    1744#   1749    1749#   1751    1751#   1753    1753#
1755    1755#   1757    1757#   1759    1759#   1761    1761#   1763    1763#   1765    1765#   1767    1767#
1773    1773#   1775    1775#   1777    1777#   1783    1783#   1785    1785#   1787    1787#   1789    1789#
1791    1791#   1793    1793#   1795    1795#   1797    1797#   1802#   1802    1804    1804#   1809#   1809
1811    1811#   1813    1813#   1818#   1818    1823#   1823    1825    1825#   1975#   1975    1977    1977#
1979    1979#   1981    1981#   1984    1984#   1986    1986#   1988    1988#   1990    1990#   1992    1992#
1994    1994#   1997    1997#   2002    2002#   2008    2008#   2010    2010#   2012    2012#   2014    2014#
2016    2016#   2018    2018#   2020    2020#   2022    2022#   2027#   2027    2029    2029#   2031    2031#
2033    2033#   2035    2035#   2037    2037#   2039    2039#   2041    2041#   2043    2043#   2045    2045#
2047    2047#   2049    2049#   2051    2051#   2053    2053#   2055    2055#   2057    2057#   2059    2059#
2061    2061#   2063    2063#   2065    2065#   2070#   2070    2072    2072#   2074    2074#   2080#   2080
2082    2082#   2084    2084#   2086    2086#   2088    2088#   2090    2090#   2092    2092#   2094    2094#
2096    2096#   2098    2098#   2100    2100#   2102    2102#   2107#   2107    2109    2109#   2269#   2269
2272    2272#   2274    2274#   2276    2276#   2278    2278#   2280    2280#   2283    2283#   2285    2285#
2287    2287#   2289    2289#   2292    2292#   2298    2298#   2300    2300#   2302    2302#   2308    2308#
2310    2310#   2312    2312#   2314    2314#   2316    2316#   2318    2318#   2320    2320#   2322    2322#
2327#   2327    2329    2329#   2331    2331#   2333    2333#   2335    2335#   2337    2337#   2339    2339#
2341    2341#   2343    2343#   2345    2345#   2347    2347#   2349    2349#   2351    2351#   2353    2353#
2355    2355#   2357    2357#   2359    2359#   2361    2361#   2363    2363#   2365    2365#   2370#   2370
2372    2372#   2374    2374#   2380#   2380    2382    2382#   2384    2384#   2386    2386#   2388    2388#
2390    2390#   2392    2392#   2394    2394#   2396    2396#   2398    2398#   2400    2400#   2402    2402#
2407#   2407    2409    2409#   2485#   2485    2487    2487#   2594#   2594    2596    2596#   2598    2598#
2600    2600#   2602    2602#   2855#   2855    2857    2857#   2859    2859#   2861    2861#   2866#   2866
```

```
2868   2868#  2870   2870#  2872   2872#  2874   2874#  2876   2876#  2881#  2881   2883   2883#   SEQ 1839
2885#  2885   2890#  2890   2892   2892#  2897#  2897   2899   2899#  2901   2901#  2903   2903#
2905   2905#  2907   2907#  2909   2909#  2911   2911#  2913   2913#  3165#  3165   3167   3167#
3169   3169#  3171   3171#  3176#  3176   3178   3178#  3180   3180#  3182   3182#  3184   3184#
3186   3186#  3191#  3191   3193   3193#  3195#  3195   3200#  3200   3202   3202#  3207#  3207
3209   3209#  3211   3211#  3213   3213#  3215   3215#  3217   3217#  3219   3219#  3221   3221#
3223   3223#  3482#  3482   3484   3484#  3486   3486#  3488   3488#  3493#  3493   3495   3495#
3497   3497#  3499   3499#  3501   3501#  3503   3503#  3508#  3508   3510   3510#  3512   3512#
3514   3514#  3516   3516#  3518   3518#  3520#  3520   3525#  3525   3527   3527#  3532#  3532
3534   3534#  3536   3536#  3538   3538#  3540   3540#  3542   3542#  3544   3544#  3546   3546#
3548   3548#  3805#  3805   3807   3807#  3809   3809#  3811   3811#  3816#  3816   3818   3818#
3820   3820#  3822   3822#  3824   3824#  3826   3826#  3831#  3831   3833   3833#  3835   3835#
3837   3837#  3839   3839#  3841   3841#  3843#  3843   3848#  3848   3850   3850#  3855#  3855
3857   3857#  3859   3859#  3861   3861#  3863   3863#  3865   3865#  3867   3867#  3869   3869#
3871   3871#  4128#  4128   4130   4130#  4132   4132#  4134   4134#  4139#  4139   4141   4141#
4143   4143#  4145   4145#  4147   4147#  4149   4149#  4154#  4154   4156   4156#  4158   4158#
4160   4160#  4162   4162#  4164   4164#  4166#  4166   4171#  4171   4173   4173#  4178#  4178
4180   4180#  4182   4182#  4184   4184#  4186   4186#  4188   4188#  4190   4190#  4192   4192#
4194   4194#  4452#  4452   4454   4454#  4456   4456#  4458   4458#  4463#  4463   4465   4465#
4467   4467#  4469   4469#  4471   4471#  4473   4473#  4478#  4478   4480   4480#  4482   4482#
4484   4484#  4486   4486#  4488   4488#  4490#  4490   4495#  4495   4497   4497#  4502#  4502
4504   4504#  4506   4506#  4508   4508#  4510   4510#  4512   4512#  4514   4514#  4516   4516#
4518   4518#  4740#  4740   4742   4742#  4744   4744#  4746   4746#  4751#  4751   4753   4753#
4755   4755#  4757   4757#  4759   4759#  4761   4761#  4766#  4766   4768   4768#  4770   4770#
4772   4772#  4774   4774#  4776   4776#  4778#  4778   4783#  4783   4785   4785#  4790#  4790
4792   4792#  4794   4794#  4796   4796#  4798   4798#  4800   4800#  4802   4802#  4804   4804#
4806   4806#  5027#  5027   5029   5029#  5031   5031#  5033   5033#  5038#  5038   5040   5040#
5042   5042#  5044   5044#  5046   5046#  5048   5048#  5053#  5053   5055   5055#  5057   5057#
5059   5059#  5061   5061#  5063   5063#  5065#  5065   5070#  5070   5072   5072#  5077#  5077
5079   5079#  5081   5081#  5083   5083#  5085   5085#  5087   5087#  5089   5089#  5091   5091#
5093   5093#  5313#  5313   5315   5315#  5317   5317#  5319   5319#  5324#  5324   5326   5326#
5328   5328#  5330   5330#  5332   5332#  5334   5334#  5339#  5339   5341   5341#  5343   5343#
5345   5345#  5347   5347#  5349   5349#  5351#  5351   5356#  5356   5358   5358#  5363#  5363
5365   5365#  5367   5367#  5369   5369#  5371   5371#  5373   5373#  5375   5375#  5377   5377#
5379   5379#  5602#  5602   5604   5604#  5606   5606#  5608   5608#  5613#  5613   5615   5615#
5617   5617#  5619   5619#  5621   5621#  5623   5623#  5628#  5628   5630   5630#  5632   5632#
5634   5634#  5636   5636#  5638   5638#  5640#  5640   5645#  5645   5647   5647#  5652#  5652
5654   5654#  5656   5656#  5658   5658#  5660   5660#  5662   5662#  5664   5664#  5666   5666#
5668   5668#  5774#  5774   5776   5776#  5778   5778#  5780   5780#  5782   5782#  6039#  6039
6041   6041#  6043   6043#  6045   6045#  6050#  6050   6052   6052#  6054   6054#  6056   6056#
6058   6058#  6060   6060#  6062   6062#  6064   6064#  6069#  6069   6071   6071#  6073#  6073
6078#  6078   6080   6080#  6085#  6085   6087   6087#  6089   6089#  6091   6091#  6093   6093#
6095   6095#  6097   6097#  6099   6099#  6101   6101#  6359#  6359   6361   6361#  6363   6363#
6365   6365#  6370#  6370   6372   6372#  6374   6374#  6376   6376#  6378   6378#  6380   6380#
6382   6382#  6384   6384#  6389#  6389   6391   6391#  6393   6393#  6395   6395#  6397   6397#
6399   6399#  6401#  6401   6406#  6406   6408   6408#  6413#  6413   6415   6415#  6417   6417#
6419   6419#  6421   6421#  6423   6423#  6425   6425#  6427   6427#  6429   6429#  6685#  6685
6687   6687#  6689   6689#  6691   6691#  6696#  6696   6698   6698#  6700   6700#  6702   6702#
6704   6704#  6706   6706#  6708   6708#  6710   6710#  6715#  6715   6717   6717#  6719   6719#
6721   6721#  6723   6723#  6725   6725#  6727#  6727   6732#  6732   6734   6734#  6739#  6739
6741   6741#  6743   6743#  6745   6745#  6747   6747#  6749   6749#  6751   6751#  6753   6753#
6755   6755#  7013#  7013   7015   7015#  7017   7017#  7019   7019#  7024#  7024   7026   7026#
7028   7028#  7030   7030#  7032   7032#  7034   7034#  7036   7036#  7038   7038#  7043#  7043
```

```
        7045    7045#   7047    7047#   7049    7049#   7051    7051#   7053    7053#   7055#   7055    7060#   7060        SEQ 1840
        7062    7062#   7067#   7067    7069    7069#   7071    7071#   7073    7073#   7075    7075#   7077    7077#
        7079    7079#   7081    7081#   7083    7083#   7341#   7341    7343    7343#   7345    7345#   7347    7347#
        7352#   7352    7354    7354#   7356    7356#   7358    7358#   7360    7360#   7362    7362#   7364    7364#
        7366    7366#   7371#   7371    7373    7373#   7375    7375#   7377    7377#   7379    7379#   7381    7381#
        7383#   7383    7388#   7388    7390    7390#   7395#   7395    7397    7397#   7399    7399#   7401    7401#
        7403    7403#   7405    7405#   7407    7407#   7409    7409#   7411    7411#   7633#   7633    7635    7635#
        7637    7637#   7639    7639#   7644#   7644    7646    7646#   7648    7648#   7650    7650#   7652    7652#
        7654    7654#   7656    7656#   7658    7658#   7663#   7663    7665    7665#   7667    7667#   7669    7669#
        7671    7671#   7673    7673#   7675#   7675    7680#   7680    7682    7682#   7687#   7687    7689    7689#
        7691    7691#   7693    7693#   7695    7695#   7697    7697#   7699    7699#   7701    7701#   7703    7703#
        7924#   7924    7926    7926#   7928    7928#   7930    7930#   7935#   7935    7937    7937#   7939    7939#
        7941    7941#   7943    7943#   7945    7945#   7947    7947#   7949    7949#   7954#   7954    7956    7956#
        7958    7958#   7960    7960#   7962    7962#   7964    7964#   7966#   7966    7971#   7971    7973    7973#
        7978#   7978    7980    7980#   7982    7982#   7984    7984#   7986    7986#   7988    7988#   7990    7990#
        7992    7992#   7994    7994#   8214#   8214    8216    8216#   8218    8218#   8220    8220#   8225#   8225
        8227    8227#   8229    8229#   8231    8231#   8233    8233#   8235    8235#   8237    8237#   8239    8239#
        8244#   8244    8246    8246#   8248    8248#   8250    8250#   8252    8252#   8254    8254#   8256#   8256
        8261#   8261    8263    8263#   8268#   8268    8270    8270#   8272    8272#   8274    8274#   8276    8276#
        8278    8278#   8280    8280#   8282    8282#   8284    8284#   8503#   8503    8505    8505#   8507    8507#
        8509    8509#   8514#   8514    8516    8516#   8518    8518#   8520    8520#   8522    8522#   8524    8524#
        8526    8526#   8528    8528#   8533#   8533    8535    8535#   8537    8537#   8539    8539#   8541    8541#
        8543    8543#   8545#   8545    8550#   8550    8552    8552#   8557#   8557    8559    8559#   8561    8561#
        8563    8563#   8565    8565#   8567    8567#   8569    8569#   8571    8571#   8573    8573#   8758#   8758
        8760    8760#   8762    8762#   8764    8764#   8766    8766#   8768    8768#   8770    8770#   8772    8772#
        8774    8774#   8776    8776#   8778    8778#   8780    8780#   8782    8782#   8784    8784#   8786    8786#
        8791#   8791    8793    8793#   8795    8795#   8797    8797#   8799    8799#   8801    8801#   8803    8803#
        8805    8805#   8810#   8810    8812    8812#   8814#   8814    8819#   8819    8821    8821#   8823    8823#
        8825    8825#   8830#   8830    8832    8832#   8837#   8837    8839    8839#   8841    8841#   8843    8843#
        8845    8845#   8847    8847#   8849    8849#   8851    8851#   8853    8853#   8858#   8858    8860    8860#
        8862    8862#   8864    8864#   8866    8866#   8868    8868#   8870    8870#   8872    8872#   8874    8874#
%ML      194#    194     196#    196     198#    198     200#    200     202#    202     204#    204     206#    206
         208#    208     210#    210     212#    212     214#    214     216#    216     218#    218     220#    220
         222#    222     224#    224     226#    226     228#    228     230#    230     232#    232     234#    234
         236#    236     238#    238     240#    240     242#    242     244#    244     246#    246     248#    248
         250#    250     252#    252     254#    254     256#    256     258#    258     260#    260     262#    262
         264#    264     266#    266     268#    268     273#    273     275#    275     280#    280     282#    282
         287#    287     289#    289     291#    291     296#    296     298#    298     300#    300     305#    305
         307#    307     309#    309     314#    314     316#    316     318#    318     323#    323     325#    325
         327#    327     332#    332     334#    334     336#    336     341#    341     343#    343     345#    345
         350#    350     352#    352     354#    354     359#    359     361#    361     363#    363     368#    368
         370#    370     372#    372     377#    377     379#    379     381#    381     386#    386     388#    388
         390#    390     395#    395     397#    397     399#    399     404#    404     406#    406     408#    408
         413#    413     415#    415     417#    417     422#    422     424#    424     426#    426     431#    431
         433#    433     435#    435     440#    440     442#    442     444#    444     449#    449     451#    451
         453#    453     458#    458     460#    460     462#    462     467#    467     469#    469     471#    471
         476#    476     478#    478     480#    480     485#    485     487#    487     489#    489     494#    494
         496#    496     498#    498     503#    503     505#    505     507#    507     512#    512     514#    514
         516#    516     521#    521     523#    523     525#    525     530#    530     532#    532     534#    534
         539#    539     541#    541     543#    543     548#    548     550#    550     552#    552     557#    557
         559#    559     561#    561     566#    566     568#    568     570#    570     575#    575     577#    577
         579#    579     584#    584     586#    586     588#    588     593#    593     595#    595     597#    597
         599#    599     604#    604     609#    609     611#    611     613#    613     615#    615     617#    617
```

```
  717#   717    719#   719    721#   721    723#   723    725#   725    727#   727    729#   729      SEQ 1841
  731#   731    733#   733    735#   735    740#   740    742#   742    747#   747    749#   749
  754#   754    756#   756    758#   758    763#   763    765#   765    767#   767    772#   772
  774#   774    776#   776    781#   781    783#   783    785#   785    790#   790    792#   792
  794#   794    799#   799    801#   801    803#   803    808#   808    810#   810    812#   812
  817#   817    822#   822    824#   824    826#   826    828#   828    830#   830    832#   832
  938#   938    943#   943    948#   948    950#   950    955#   955    957#   957    959#   959
  964#   964    966#   966    971#   971    973#   973   1062#  1062   1064#  1064   1066#  1066
 1071#  1071   1076#  1076   1081#  1081   1083#  1083   1088#  1088   1093#  1093   1095#  1095
 1100#  1100   1102#  1102   1108#  1108   1110#  1110   1112#  1112   1114#  1114   1116#  1116
 1118#  1118   1120#  1120   1122#  1122   1127#  1127   1129#  1129   1330#  1330   1332#  1332
 1334#  1334   1336#  1336   1341#  1341   1343#  1343   1345#  1345   1347#  1347   1349#  1349
 1351#  1351   1356#  1356   1358#  1358   1360#  1360   1365#  1365   1371#  1371   1373#  1373
 1375#  1375   1377#  1377   1379#  1379   1381#  1381   1383#  1383   1385#  1385   1392#  1392
 1394#  1394   1396#  1396   1398#  1398   1400#  1400   1402#  1402   1404#  1404   1406#  1406
 1408#  1408   1410#  1410   1412#  1412   1414#  1414   1416#  1416   1421#  1421   1423#  1423
 1428#  1428   1430#  1430   1432#  1432   1437#  1437   1439#  1439   1444#  1444   1446#  1446
 1451#  1451   1456#  1456   1458#  1458   1586#  1586   1588#  1588   1590#  1590   1592#  1592
 1676#  1676   1678#  1678   1680#  1680   1682#  1682   1684#  1684   1686#  1686   1691#  1691
 1693#  1693   1695#  1695   1697#  1697   1699#  1699   1701#  1701   1706#  1706   1708#  1708
 1710#  1710   1712#  1712   1714#  1714   1716#  1716   1718#  1718   1720#  1720   1722#  1722
 1724#  1724   1726#  1726   1728#  1728   1730#  1730   1732#  1732   1734#  1734   1736#  1736
 1738#  1738   1740#  1740   1742#  1742   1744#  1744   1749#  1749   1751#  1751   1753#  1753
 1755#  1755   1757#  1757   1759#  1759   1761#  1761   1763#  1763   1765#  1765   1767#  1767
 1773#  1773   1775#  1775   1777#  1777   1783#  1783   1785#  1785   1787#  1787   1789#  1789
 1791#  1791   1793#  1793   1795#  1795   1797#  1797   1802#  1802   1804#  1804   1809#  1809
 1811#  1811   1813#  1813   1818#  1818   1823#  1823   1825#  1825   1975#  1975   1977#  1977
 1979#  1979   1981#  1981   1984#  1984   1986#  1986   1988#  1988   1990#  1990   1992#  1992
 1994#  1994   1997#  1997   2002#  2002   2008#  2008   2010#  2010   2012#  2012   2014#  2014
 2016#  2016   2018#  2018   2020#  2020   2022#  2022   2027#  2027   2029#  2029   2031#  2031
 2033#  2033   2035#  2035   2037#  2037   2039#  2039   2041#  2041   2043#  2043   2045#  2045
 2047#  2047   2049#  2049   2051#  2051   2053#  2053   2055#  2055   2057#  2057   2059#  2059
 2061#  2061   2063#  2063   2065#  2065   2070#  2070   2072#  2072   2074#  2074   2080#  2080
 2082#  2082   2084#  2084   2086#  2086   2088#  2088   2090#  2090   2092#  2092   2094#  2094
 2096#  2096   2098#  2098   2100#  2100   2102#  2102   2107#  2107   2109#  2109   2269#  2269
 2272#  2272   2274#  2274   2276#  2276   2278#  2278   2280#  2280   2283#  2283   2285#  2285
 2287#  2287   2289#  2289   2292#  2292   2298#  2298   2300#  2300   2302#  2302   2308#  2308
 2310#  2310   2312#  2312   2314#  2314   2316#  2316   2318#  2318   2320#  2320   2322#  2322
 2327#  2327   2329#  2329   2331#  2331   2333#  2333   2335#  2335   2337#  2337   2339#  2339
 2341#  2341   2343#  2343   2345#  2345   2347#  2347   2349#  2349   2351#  2351   2353#  2353
 2355#  2355   2357#  2357   2359#  2359   2361#  2361   2363#  2363   2365#  2365   2370#  2370
 2372#  2372   2374#  2374   2380#  2380   2382#  2382   2384#  2384   2386#  2386   2388#  2388
 2390#  2390   2392#  2392   2394#  2394   2396#  2396   2398#  2398   2400#  2400   2402#  2402
 2407#  2407   2409#  2409   2485#  2485   2487#  2487   2594#  2594   2596#  2596   2598#  2598
 2600#  2600   2602#  2602   2855#  2855   2857#  2857   2859#  2859   2861#  2861   2866#  2866
 2868#  2868   2870#  2870   2872#  2872   2874#  2874   2876#  2876   2881#  2881   2883#  2883
 2885#  2885   2890#  2890   2892#  2892   2897#  2897   2899#  2899   2901#  2901   2903#  2903
 2905#  2905   2907#  2907   2909#  2909   2911#  2911   2913#  2913   3165#  3165   3167#  3167
 3169#  3169   3171#  3171   3176#  3176   3178#  3178   3180#  3180   3182#  3182   3184#  3184
 3186#  3186   3191#  3191   3193#  3193   3195#  3195   3200#  3200   3202#  3202   3207#  3207
 3209#  3209   3211#  3211   3213#  3213   3215#  3215   3217#  3217   3219#  3219   3221#  3221
 3223#  3223   3482#  3482   3484#  3484   3486#  3486   3488#  3488   3493#  3493   3495#  3495
 3497#  3497   3499#  3499   3501#  3501   3503#  3503   3508#  3508   3510#  3510   3512#  3512
```

EQ 1829

```
3514#  3514   3516#  3516   3518#  3518   3520#  3520   3525#  3525   3527#  3527   3532#  3532
3534#  3534   3536#  3536   3538#  3538   3540#  3540   3542#  3542   3544#  3544   3546#  3546
3548#  3548   3805#  3805   3807#  3807   3809#  3809   3811#  3811   3816#  3816   3818#  3818
3820#  3820   3822#  3822   3824#  3824   3826#  3826   3831#  3831   3833#  3833   3835#  3835
3837#  3837   3839#  3839   3841#  3841   3843#  3843   3848#  3848   3850#  3850   3855#  3855
3857#  3857   3859#  3859   3861#  3861   3863#  3863   3865#  3865   3867#  3867   3869#  3869
3871#  3871   4128#  4128   4130#  4130   4132#  4132   4134#  4134   4139#  4139   4141#  4141
4143#  4143   4145#  4145   4147#  4147   4149#  4149   4154#  4154   4156#  4156   4158#  4158
4160#  4160   4162#  4162   4164#  4164   4166#  4166   4171#  4171   4173#  4173   4178#  4178
4180#  4180   4182#  4182   4184#  4184   4186#  4186   4188#  4188   4190#  4190   4192#  4192
4194#  4194   4452#  4452   4454#  4454   4456#  4456   4458#  4458   4463#  4463   4465#  4465
4467#  4467   4469#  4469   4471#  4471   4473#  4473   4478#  4478   4480#  4480   4482#  4482
4484#  4484   4486#  4486   4488#  4488   4490#  4490   4495#  4495   4497#  4497   4502#  4502
4504#  4504   4506#  4506   4508#  4508   4510#  4510   4512#  4512   4514#  4514   4516#  4516
4518#  4518   4740#  4740   4742#  4742   4744#  4744   4746#  4746   4751#  4751   4753#  4753
4755#  4755   4757#  4757   4759#  4759   4761#  4761   4766#  4766   4768#  4768   4770#  4770
4772#  4772   4774#  4774   4776#  4776   4778#  4778   4783#  4783   4785#  4785   4790#  4790
4792#  4792   4794#  4794   4796#  4796   4798#  4798   4800#  4800   4802#  4802   4804#  4804
4806#  4806   5027#  5027   5029#  5029   5031#  5031   5033#  5033   5038#  5038   5040#  5040
5042#  5042   5044#  5044   5046#  5046   5048#  5048   5053#  5053   5055#  5055   5057#  5057
5059#  5059   5061#  5061   5063#  5063   5065#  5065   5070#  5070   5072#  5072   5077#  5077
5079#  5079   5081#  5081   5083#  5083   5085#  5085   5087#  5087   5089#  5089   5091#  5091
5093#  5093   5313#  5313   5315#  5315   5317#  5317   5319#  5319   5324#  5324   5326#  5326
5328#  5328   5330#  5330   5332#  5332   5334#  5334   5339#  5339   5341#  5341   5343#  5343
5345#  5345   5347#  5347   5349#  5349   5351#  5351   5356#  5356   5358#  5358   5363#  5363
5365#  5365   5367#  5367   5369#  5369   5371#  5371   5373#  5373   5375#  5375   5377#  5377
5379#  5379   5602#  5602   5604#  5604   5606#  5606   5608#  5608   5613#  5613   5615#  5615
5617#  5617   5619#  5619   5621#  5621   5623#  5623   5628#  5628   5630#  5630   5632#  5632
5634#  5634   5636#  5636   5638#  5638   5640#  5640   5645#  5645   5647#  5647   5652#  5652
5654#  5654   5656#  5656   5658#  5658   5660#  5660   5662#  5662   5664#  5664   5666#  5666
5668#  5668   5774#  5774   5776#  5776   5778#  5778   5780#  5780   5782#  5782   6039#  6039
6041#  6041   6043#  6043   6045#  6045   6050#  6050   6052#  6052   6054#  6054   6056#  6056
6058#  6058   6060#  6060   6062#  6062   6064#  6064   6069#  6069   6071#  6071   6073#  6073
6078#  6078   6080#  6080   6085#  6085   6087#  6087   6089#  6089   6091#  6091   6093#  6093
6095#  6095   6097#  6097   6099#  6099   6101#  6101   6359#  6359   6361#  6361   6363#  6363
6365#  6365   6370#  6370   6372#  6372   6374#  6374   6376#  6376   6378#  6378   6380#  6380
6382#  6382   6384#  6384   6389#  6389   6391#  6391   6393#  6393   6395#  6395   6397#  6397
6399#  6399   6401#  6401   6406#  6406   6408#  6408   6413#  6413   6415#  6415   6417#  6417
6419#  6419   6421#  6421   6423#  6423   6425#  6425   6427#  6427   6429#  6429   6685#  6685
6687#  6687   6689#  6689   6691#  6691   6696#  6696   6698#  6698   6700#  6700   6702#  6702
6704#  6704   6706#  6706   6708#  6708   6710#  6710   6715#  6715   6717#  6717   6719#  6719
6721#  6721   6723#  6723   6725#  6725   6727#  6727   6732#  6732   6734#  6734   6739#  6739
6741#  6741   6743#  6743   6745#  6745   6747#  6747   6749#  6749   6751#  6751   6753#  6753
6755#  6755   7013#  7013   7015#  7015   7017#  7017   7019#  7019   7024#  7024   7026#  7026
7028#  7028   7030#  7030   7032#  7032   7034#  7034   7036#  7036   7038#  7038   7043#  7043
7045#  7045   7047#  7047   7049#  7049   7051#  7051   7053#  7053   7055#  7055   7060#  7060
7062#  7062   7067#  7067   7069#  7069   7071#  7071   7073#  7073   7075#  7075   7077#  7077
7079#  7079   7081#  7081   7083#  7083   7341#  7341   7343#  7343   7345#  7345   7347#  7347
7352#  7352   7354#  7354   7356#  7356   7358#  7358   7360#  7360   7362#  7362   7364#  7364
7366#  7366   7371#  7371   7373#  7373   7375#  7375   7377#  7377   7379#  7379   7381#  7381
7383#  7383   7388#  7388   7390#  7390   7395#  7395   7397#  7397   7399#  7399   7401#  7401
7403#  7403   7405#  7405   7407#  7407   7409#  7409   7411#  7411   7633#  7633   7635#  7635
7637#  7637   7639#  7639   7644#  7644   7646#  7646   7648#  7648   7650#  7650   7652#  7652
```

SEQ 1842

```
EQ 1830   7654#   7654   7656#   7656   7658#   7658   7663#   7663   7665#   7665   7667#   7667   7669#   7669
          7671#   7671   7673#   7673   7675#   7675   7680#   7680   7682#   7682   7687#   7687   7689#   7689     SEQ 1843
          7691#   7691   7693#   7693   7695#   7695   7697#   7697   7699#   7699   7701#   7701   7703#   7703
          7924#   7924   7926#   7926   7928#   7928   7930#   7930   7935#   7935   7937#   7937   7939#   7939
          7941#   7941   7943#   7943   7945#   7945   7947#   7947   7949#   7949   7954#   7954   7956#   7956
          7958#   7958   7960#   7960   7962#   7962   7964#   7964   7966#   7966   7971#   7971   7973#   7973
          7978#   7978   7980#   7980   7982#   7982   7984#   7984   7986#   7986   7988#   7988   7990#   7990
          7992#   7992   7994#   7994   8214#   8214   8216#   8216   8218#   8218   8220#   8220   8225#   8225
          8227#   8227   8229#   8229   8231#   8231   8233#   8233   8235#   8235   8237#   8237   8239#   8239
          8244#   8244   8246#   8246   8248#   8248   8250#   8250   8252#   8252   8254#   8254   8256#   8256
          8261#   8261   8263#   8263   8268#   8268   8270#   8270   8272#   8272   8274#   8274   8276#   8276
          8278#   8278   8280#   8280   8282#   8282   8284#   8284   8503#   8503   8505#   8505   8507#   8507
          8509#   8509   8514#   8514   8516#   8516   8518#   8518   8520#   8520   8522#   8522   8524#   8524
          8526#   8526   8528#   8528   8533#   8533   8535#   8535   8537#   8537   8539#   8539   8541#   8541             .M
          8543#   8543   8545#   8545   8550#   8550   8552#   8552   8557#   8557   8559#   8559   8561#   8561
          8563#   8563   8565#   8565   8567#   8567   8569#   8569   8571#   8571   8573#   8573   8758#   8758
          8760#   8760   8762#   8762   8764#   8764   8766#   8766   8768#   8768   8770#   8770   8772#   8772
          8774#   8774   8776#   8776   8778#   8778   8780#   8780   8782#   8782   8784#   8784   8786#   8786
          8791#   8791   8793#   8793   8795#   8795   8797#   8797   8799#   8799   8801#   8801   8803#   8803
          8805#   8805   8810#   8810   8812#   8812   8814#   8814   8819#   8819   8821#   8821   8823#   8823
          8825#   8825   8830#   8830   8832#   8832   8837#   8837   8839#   8839   8841#   8841   8843#   8843
          8845#   8845   8847#   8847   8849#   8849   8851#   8851   8853#   8853   8858#   8858   8860#   8860
          8862#   8862   8864#   8864   8866#   8866   8868#   8868   8870#   8870   8872#   8872   8874#   8874
%MR        194#    194    196#    196    198#    198    200#    200    202#    202    204#    204    206#    206
           208#    208    210#    210    212#    212    214#    214    216#    216    218#    218    220#    220
           222#    222    224#    224    226#    226    228#    228    230#    230    232#    232    234#    234
           236#    236    238#    238    240#    240    242#    242    244#    244    246#    246    248#    248
           250#    250    252#    252    254#    254    256#    256    258#    258    260#    260    262#    262
           264#    264    266#    266    268#    268    273#    273    275#    275    280#    280    282#    282
           287#    287    289#    289    291#    291    296#    296    298#    298    300#    300    305#    305
           307#    307    309#    309    314#    314    316#    316    318#    318    323#    323    325#    325
           327#    327    332#    332    334#    334    336#    336    341#    341    343#    343    345#    345
           350#    350    352#    352    354#    354    359#    359    361#    361    363#    363    368#    368
           370#    370    372#    372    377#    377    379#    379    381#    381    386#    386    388#    388
           390#    390    395#    395    397#    397    399#    399    404#    404    406#    406    408#    408
           413#    413    415#    415    417#    417    422#    422    424#    424    426#    426    431#    431
           433#    433    435#    435    440#    440    442#    442    444#    444    449#    449    451#    451
           453#    453    458#    458    460#    460    462#    462    467#    467    469#    469    471#    471
           476#    476    478#    478    480#    480    485#    485    487#    487    489#    489    494#    494
           496#    496    498#    498    503#    503    505#    505    507#    507    512#    512    514#    514
           516#    516    521#    521    523#    523    525#    525    530#    530    532#    532    534#    534
           539#    539    541#    541    543#    543    548#    548    550#    550    552#    552    557#    557
           559#    559    561#    561    566#    566    568#    568    570#    570    575#    575    577#    577
           579#    579    584#    584    586#    586    588#    588    593#    593    595#    595    597#    597
           599#    599    604#    604    609#    609    611#    611    613#    613    615#    615    617#    617
           717#    717    719#    719    721#    721    723#    723    725#    725    727#    727    729#    729
           731#    731    733#    733    735#    735    740#    740    742#    742    747#    747    749#    749
           754#    754    756#    756    758#    758    763#    763    765#    765    767#    767    772#    772
           774#    774    776#    776    781#    781    783#    783    785#    785    790#    790    792#    792
           794#    794    799#    799    801#    801    803#    803    808#    808    810#    810    812#    812
           817#    817    822#    822    824#    824    826#    826    828#    828    830#    830    832#    832
           938#    938    943#    943    948#    948    950#    950    955#    955    957#    957    959#    959
           964#    964    966#    966    971#    971    973#    973   1062#   1062   1064#   1064   1066#   1066
```

```
1071#  1071   1076#  1076   1081#  1081   1083#  1083   1088#  1088   1093#  1093   1095#  1095      SEQ 1844
1100#  1100   1102#  1102   1108#  1108   1110#  1110   1112#  1112   1114#  1114   1116#  1116
1118#  1118   1120#  1120   1122#  1122   1127#  1127   1129#  1129   1330#  1330   1332#  1332
1334#  1334   1336#  1336   1341#  1341   1343#  1343   1345#  1345   1347#  1347   1349#  1349
1351#  1351   1356#  1356   1358#  1358   1360#  1360   1365#  1365   1371#  1371   1373#  1373
1375#  1375   1377#  1377   1379#  1379   1381#  1381   1383#  1383   1385#  1385   1392#  1392
1394#  1394   1396#  1396   1398#  1398   1400#  1400   1402#  1402   1404#  1404   1406#  1406
1408#  1408   1410#  1410   1412#  1412   1414#  1414   1416#  1416   1421#  1421   1423#  1423
1428#  1428   1430#  1430   1432#  1432   1437#  1437   1439#  1439   1444#  1444   1446#  1446
1451#  1451   1456#  1456   1458#  1458   1586#  1586   1588#  1588   1590#  1590   1592#  1592
1676#  1676   1678#  1678   1680#  1680   1682#  1682   1684#  1684   1686#  1686   1691#  1691
1693#  1693   1695#  1695   1697#  1697   1699#  1699   1701#  1701   1706#  1706   1708#  1708
1710#  1710   1712#  1712   1714#  1714   1716#  1716   1718#  1718   1720#  1720   1722#  1722
1724#  1724   1726#  1726   1728#  1728   1730#  1730   1732#  1732   1734#  1734   1736#  1736
1738#  1738   1740#  1740   1742#  1742   1744#  1744   1749#  1749   1751#  1751   1753#  1753
1755#  1755   1757#  1757   1759#  1759   1761#  1761   1763#  1763   1765#  1765   1767#  1767
1773#  1773   1775#  1775   1777#  1777   1783#  1783   1785#  1785   1787#  1787   1789#  1789
1791#  1791   1793#  1793   1795#  1795   1797#  1797   1802#  1802   1804#  1804   1809#  1809
1811#  1811   1813#  1813   1818#  1818   1823#  1823   1825#  1825   1975#  1975   1977#  1977
1979#  1979   1981#  1981   1984#  1984   1986#  1986   1988#  1988   1990#  1990   1992#  1992
1994#  1994   1997#  1997   2002#  2002   2008#  2008   2010#  2010   2012#  2012   2014#  2014
2016#  2016   2018#  2018   2020#  2020   2022#  2022   2027#  2027   2029#  2029   2031#  2031
2033#  2033   2035#  2035   2037#  2037   2039#  2039   2041#  2041   2043#  2043   2045#  2045
2047#  2047   2049#  2049   2051#  2051   2053#  2053   2055#  2055   2057#  2057   2059#  2059
2061#  2061   2063#  2063   2065#  2065   2070#  2070   2072#  2072   2074#  2074   2080#  2080
2082#  2082   2084#  2084   2086#  2086   2088#  2088   2090#  2090   2092#  2092   2094#  2094
2096#  2096   2098#  2098   2100#  2100   2102#  2102   2107#  2107   2109#  2109   2269#  2269
2272#  2272   2274#  2274   2276#  2276   2278#  2278   2280#  2280   2283#  2283   2285#  2285
2287#  2287   2289#  2289   2292#  2292   2298#  2298   2300#  2300   2302#  2302   2308#  2308
2310#  2310   2312#  2312   2314#  2314   2316#  2316   2318#  2318   2320#  2320   2322#  2322
2327#  2327   2329#  2329   2331#  2331   2333#  2333   2335#  2335   2337#  2337   2339#  2339
2341#  2341   2343#  2343   2345#  2345   2347#  2347   2349#  2349   2351#  2351   2353#  2353
2355#  2355   2357#  2357   2359#  2359   2361#  2361   2363#  2363   2365#  2365   2370#  2370
2372#  2372   2374#  2374   2380#  2380   2382#  2382   2384#  2384   2386#  2386   2388#  2388
2390#  2390   2392#  2392   2394#  2394   2396#  2396   2398#  2398   2400#  2400   2402#  2402
2407#  2407   2409#  2409   2485#  2485   2487#  2487   2594#  2594   2596#  2596   2598#  2598
2600#  2600   2602#  2602   2855#  2855   2857#  2857   2859#  2859   2861#  2861   2866#  2866
2868#  2868   2870#  2870   2872#  2872   2874#  2874   2876#  2876   2881#  2881   2883#  2883
2885#  2885   2890#  2890   2892#  2892   2897#  2897   2899#  2899   2901#  2901   2903#  2903
2905#  2905   2907#  2907   2909#  2909   2911#  2911   2913#  2913   3165#  3165   3167#  3167
3169#  3169   3171#  3171   3176#  3176   3178#  3178   3180#  3180   3182#  3182   3184#  3184
3186#  3186   3191#  3191   3193#  3193   3195#  3195   3200#  3200   3202#  3202   3207#  3207
3209#  3209   3211#  3211   3213#  3213   3215#  3215   3217#  3217   3219#  3219   3221#  3221
3223#  3223   3482#  3482   3484#  3484   3486#  3486   3488#  3488   3493#  3493   3495#  3495
3497#  3497   3499#  3499   3501#  3501   3503#  3503   3508#  3508   3510#  3510   3512#  3512
3514#  3514   3516#  3516   3518#  3518   3520#  3520   3525#  3525   3527#  3527   3532#  3532
3534#  3534   3536#  3536   3538#  3538   3540#  3540   3542#  3542   3544#  3544   3546#  3546
3548#  3548   3805#  3805   3807#  3807   3[illegible]9#  3809   3811#  3811   3816#  3816   3818#  3818
3820#  3820   3822#  3822   3824#  3824   3[illegible]#  3826   3831#  3831   3833#  3833   3835#  3835
3837#  3837   3839#  3839   3841#  3841   3843#  3843   3848#  3848   3850#  3850   3855#  3855
3857#  3857   3859#  3859   3861#  3861   3863#  3863   3865#  3865   3867#  3867   3869#  3869
3871#  3871   4128#  4128   4130#  4130   4132#  4132   4134#  4134   4139#  4139   4141#  4141
4143#  4143   4145#  4145   4147#  4147   4149#  4149   4154#  4154   4156#  4156   4158#  4158
```

```
Q 1832
     4160#  4160    4162#  4162    4164#  4164    4166#  4166    4171#  4171    4173#  4173    4178#  4178
     4180#  4180    4182#  4182    4184#  4184    4186#  4186    4188#  4188    4190#  4190    4192#  4192     SEQ 1845
     4194#  4194    4452#  4452    4454#  4454    4456#  4456    4458#  4458    4463#  4463    4465#  4465
     4467#  4467    4469#  4469    4471#  4471    4473#  4473    4478#  4478    4480#  4480    4482#  4482
     4484#  4484    4486#  4486    4488#  4488    4490#  4490    4495#  4495    4497#  4497    4502#  4502
     4504#  4504    4506#  4506    4508#  4508    4510#  4510    4512#  4512    4514#  4514    4516#  4516
     4518#  4518    4740#  4740    4742#  4742    4744#  4744    4746#  4746    4751#  4751    4753#  4753
     4755#  4755    4757#  4757    4759#  4759    4761#  4761    4766#  4766    4768#  4768    4770#  4770
     4772#  4772    4774#  4774    4776#  4776    4778#  4778    4783#  4783    4785#  4785    4790#  4790
     4792#  4792    4794#  4794    4796#  4796    4798#  4798    4800#  4800    4802#  4802    4804#  4804
     4806#  4806    5027#  5027    5029#  5029    5031#  5031    5033#  5033    5038#  5038    5040#  5040
     5042#  5042    5044#  5044    5046#  5046    5048#  5048    5053#  5053    5055#  5055    5057#  5057
     5059#  5059    5061#  5061    5063#  5063    5065#  5065    5070#  5070    5072#  5072    5077#  5077
     5079#  5079    5081#  5081    5083#  5083    5085#  5085    5087#  5087    5089#  5089    5091#  5091
     5093#  5093    5313#  5313    5315#  5315    5317#  5317    5319#  5319    5324#  5324    5326#  5326
     5328#  5328    5330#  5330    5332#  5332    5334#  5334    5339#  5339    5341#  5341    5343#  5343
     5345#  5345    5347#  5347    5349#  5349    5351#  5351    5356#  5356    5358#  5358    5363#  5363
     5365#  5365    5367#  5367    5369#  5369    5371#  5371    5373#  5373    5375#  5375    5377#  5377
     5379#  5379    5602#  5602    5604#  5604    5606#  5606    5608#  5608    5613#  5613    5615#  5615
     5617#  5617    5619#  5619    5621#  5621    5623#  5623    5628#  5628    5630#  5630    5632#  5632
     5634#  5634    5636#  5636    5638#  5638    5640#  5640    5645#  5645    5647#  5647    5652#  5652
     5654#  5654    5656#  5656    5658#  5658    5660#  5660    5662#  5662    5664#  5664    5666#  5666
     5668#  5668    5774#  5774    5776#  5776    5778#  5778    5780#  5780    5782#  5782    6039#  6039
     6041#  6041    6043#  6043    6045#  6045    6050#  6050    6052#  6052    6054#  6054    6056#  6056
     6058#  6058    6060#  6060    6062#  6062    6064#  6064    6069#  6069    6071#  6071    6073#  6073
     6078#  6078    6080#  6080    6085#  6085    6087#  6087    6089#  6089    6091#  6091    6093#  6093
     6095#  6095    6097#  6097    6099#  6099    6101#  6101    6359#  6359    6361#  6361    6363#  6363
     6365#  6365    6370#  6370    6372#  6372    6374#  6374    6376#  6376    6378#  6378    6380#  6380
     6382#  6382    6384#  6384    6389#  6389    6391#  6391    6393#  6393    6395#  6395    6397#  6397
     6399#  6399    6401#  6401    6406#  6406    6408#  6408    6413#  6413    6415#  6415    6417#  6417
     6419#  6419    6421#  6421    6423#  6423    6425#  6425    6427#  6427    6429#  6429    6685#  6685
     6687#  6687    6689#  6689    6691#  6691    6696#  6696    6698#  6698    6700#  6700    6702#  6702
     6704#  6704    6706#  6706    6708#  6708    6710#  6710    6715#  6715    6717#  6717    6719#  6719
     6721#  6721    6723#  6723    6725#  6725    6727#  6727    6732#  6732    6734#  6734    6739#  6739
     6741#  6741    6743#  6743    6745#  6745    6747#  6747    6749#  6749    6751#  6751    6753#  6753
     6755#  6755    7013#  7013    7015#  7015    7017#  7017    7019#  7019    7024#  7024    7026#  7026
     7028#  7028    7030#  7030    7032#  7032    7034#  7034    7036#  7036    7038#  7038    7043#  7043
     7045#  7045    7047#  7047    7049#  7049    7051#  7051    7053#  7053    7055#  7055    7060#  7060
     7062#  7062    7067#  7067    7069#  7069    7071#  7071    7073#  7073    7075#  7075    7077#  7077
     7079#  7079    7081#  7081    7083#  7083    7341#  7341    7343#  7343    7345#  7345    7347#  7347
     7352#  7352    7354#  7354    7356#  7356    7358#  7358    7360#  7360    7362#  7362    7364#  7364
     7366#  7366    7371#  7371    7373#  7373    7375#  7375    7377#  7377    7379#  7379    7381#  7381
     7383#  7383    7388#  7388    7390#  7390    7395#  7395    7397#  7397    7399#  7399    7401#  7401
     7403#  7403    7405#  7405    7407#  7407    7409#  7409    7411#  7411    7633#  7633    7635#  7635
     7637#  7637    7639#  7639    7644#  7644    7646#  7646    7648#  7648    7650#  7650    7652#  7652
     7654#  7654    7656#  7656    7658#  7658    7663#  7663    7665#  7665    7667#  7667    7669#  7669
     7671#  7671    7673#  7673    7675#  7675    7680#  7680    7682#  7682    7687#  7687    7689#  7689
     7691#  7691    7693#  7693    7695#  7695    7697#  7697    7699#  7699    7701#  7701    7703#  7703
     7924#  7924    7926#  7926    7928#  7928    7930#  7930    7935#  7935    7937#  7937    7939#  7939
     7941#  7941    7943#  7943    7945#  7945    7947#  7947    7949#  7949    7954#  7954    7956#  7956
     7958#  7958    7960#  7960    7962#  7962    7964#  7964    7966#  7966    7971#  7971    7973#  7973
     7978#  7978    7980#  7980    7982#  7982    7984#  7984    7986#  7986    7988#  7988    7990#  7990
     7992#  7992    7994#  7994    8214#  8214    8216#  8216    8218#  8218    8220#  8220    8225#  8225
```

```
          8227#  8227  8229#  8229  8231#  8231  8233#  8233  8235#  8235  8237#  8237  8239#  8239     SEQ 1846
          8244#  8244  8246#  8246  8248#  8248  8250#  8250  8252#  8252  8254#  8254  8256#  8256
          8261#  8261  8263#  8263  8268#  8268  8270#  8270  8272#  8272  8274#  8274  8276#  8276
          8278#  8278  8280#  8280  8282#  8282  8284#  8284  8503#  8503  8505#  8505  8507#  8507
          8509#  8509  8514#  8514  8516#  8516  8518#  8518  8520#  8520  8522#  8522  8524#  8524
          8526#  8526  8528#  8528  8533#  8533  8535#  8535  8537#  8537  8539#  8539  8541#  8541
          8543#  8543  8545#  8545  8550#  8550  8552#  8552  8557#  8557  8559#  8559  8561#  8561
          8563#  8563  8565#  8565  8567#  8567  8569#  8569  8571#  8571  8573#  8573  8758#  8758
          8760#  8760  8762#  8762  8764#  8764  8766#  8766  8768#  8768  8770#  8770  8772#  8772
          8774#  8774  8776#  8776  8778#  8778  8780#  8780  8782#  8782  8784#  8784  8786#  8786
          8791#  8791  8793#  8793  8795#  8795  8797#  8797  8799#  8799  8801#  8801  8803#  8803
          8805#  8805  8810#  8810  8812#  8812  8814#  8814  8819#  8819  8821#  8821  8823#  8823
          8825#  8825  8830#  8830  8832#  8832  8837#  8837  8839#  8839  8841#  8841  8843#  8843
          8845#  8845  8847#  8847  8849#  8849  8851#  8851  8853#  8853  8858#  8858  8860#  8860
          8862#  8862  8864#  8864  8866#  8866  8868#  8868  8870#  8870  8872#  8872  8874#  8874
.LA        275    282   289    298   307    316   325    334   343    352   361    370   379    388
           397    406   415    424   433    442   451    460   469    478   487    496   505    514
           523    532   541    550   559    568   577    586   595    597   599    609   611    615
           742    749   756    765   774    783   792    801   810    822   824    830  1112   1118
          1120   1332  1334   1336  1343   1358  1375   1381  1383   1430  1437   1444  1740   1742
          1744   1761  1787   1793  1795   1802  1984   1992  2012   2018  2020   2072  2094   2272
          2312   2318  2320   2372  2394   2903  2905   2911  2913   3213  3215   3221  3223   3538
          3540   3546  3548   3861  3863   3869  3871   4184  4186   4192  4194   4508  4510   4516
          4518   4796  4798   4804  4806   5083  5085   5091  5093   5369  5371   5377  5379   5658
          5660   5666  5668   6091  6093   6099  6101   6419  6421   6427  6429   6745  6747   6753
          6755   7073  7075   7081  7083   7401  7403   7409  7411   7693  7695   7701  7703   7984
          7986   7992  7994   8274  8276   8282  8284   8563  8565   8571  8573   8780  8805   8819
          8823   8843  8845   8851  8853   8864  8866   8872  8874
.LADDR     194    273   280    287   296    305   314    323   332    341   350    359   368    377
           386    395   404    413   422    431   440    449   458    467   476    485   494    503
           512    521   530    539   548    557   566    575   584    593   604    609   717    740
           747    754   763    772   781    790   799    808   817    822   938    971  1062   1127
          1330   1392  1437   1444  1451   1456  1586   1676  1691   1706  1802   1809  1818   1823
          1975   2027  2070   2080  2107   2269  2327   2370  2380   2407  2485   2594  2855   2866
          2881   2885  2890   2897  3165   3176  3191   3195  3200   3207  3482   3493  3508   3520
          3525   3532  3805   3816  3831   3843  3848   3855  4128   4139  4154   4166  4171   4178
          4452   4463  4478   4490  4495   4502  4740   4751  4766   4778  4783   4790  5027   5038
          5053   5065  5070   5077  5313   5324  5339   5351  5356   5363  5602   5613  5628   5640
          5645   5652  5774   6039  6050   6069  6073   6078  6085   6359  6370   6389  6401   6406
          6413   6665  6696   6715  6727   6732  6739   7013  7024   7043  7055   7060  7067   7341
          7352   7371  7383   7388  7395   7633  7644   7663  7675   7680  7687   7924  7935   7954
          7966   7971  7978   8214  8225   8244  8256   8261  8268   8503  8514   8533  8545   8550
          8557   8758  8791   8810  8814   8819  8830   8837  8858
.LAND      273    280   287    296   305    314   323    332   341    350   359    368   377    386
           395    404   413    422   431    440   449    458   467    476   485    494   503    512
           521    530   539    548   557    566   575    584   593    604   740    747   754    763
           772    781   790    799   808    817   938   1062  1064   1334  1404   1410  1676   1691
          2102   2269  2402   2897  3207   3532  3855   4178  4502   4790  5077   5363  5652   6085
          6413   6739  7067   7395  7687   7978  8268   8557  8758   8837  8858
.LB        273    275   280    282   287    289   296    298   305    307   314    316   323    325
           332    334   341    343   350    352   359    361   368    370   377    379   386    388
           395    397   404    406   413    415   422    424   431    433   440    442   449    451
           458    460   467    469   476    478   485    487   494    496   503    505   512    514
```

```
             521     523     530     532     539     541     548     550     557     559     566     568     575     577
             584     586     593     595     597     599     604     609     613     740     742     747     749     754
             756     763     765     772     774     781     783     790     792     799     801     808     810     817
             822     828    1108    1110    1332    1334    1336    1343    1347    1349    1371    1373    1404    1406
            1408    1410    1588    1590    1592    1676    1678    1680    1684    1691    1693    1695    1699    1708
            1710    1712    1716    1720    1724    1728    1732    1736    1738    1740    1742    1744    1763    1765
            1783    1785    1981    1984    1986    1994    2008    2010    2072    2092    2094    2096    2098    2102
            2269    2272    2274    2280    2287    2292    2308    2310    2372    2392    2394    2396    2398    2402
            2859    2874    2897    2899    2903    3169    3184    3207    3209    3213    3486    3501    3532    3534
            3538    3809    3824    3855    3857    3861    4132    4147    4178    4180    4184    4456    4471    4502
            4504    4508    4744    4759    4790    4792    4796    5031    5046    5077    5079    5083    5317    5332
            5363    5365    5369    5606    5621    5652    5654    5658    6043    6062    6085    6087    6091    6363
            6382    6413    6415    6419    6689    6708    6739    6741    6745    7017    7036    7067    7069    7073
            7345    7364    7395    7397    7401    7637    7656    7687    7689    7693    7928    7947    7978    7980
            7984    8218    8237    8268    8270    8274    8507    8526    8557    8559    8563    8758    8762    8778
            8780    8803    8805    8837    8839    8843    8858    8860    8864
.LBAD       2602    5782
.LCCCA       948    1081    1093    1398    1755    1975    2086    2283    2386    2868    3178    3495    3818    4141
            4465    4753    5040    5326    5615    6052    6372    6698    7026    7354    7646    7937    8227    8516
            8793
.LCCCE      1412    1751
.LCCCP      1414    1753
.LCCER      1988    2276
.LCCFZ      1343    1765    2598    5778    8780
.LCCGC      1114    1377    1789    2014    2314    2907    3217    3542    3865    4188    4512    4800    5087    5373
            5662    6095    6423    6749    7077    7405    7697    7988    8278    8567    8847    8868
.LCCLW      1408
.LCCMB      1349
.LCENA       948    1081    1093    1114    1343    1349    1377    1398    1408    1412    1414    1751    1753    1755
            1765    1789    1975    1988    2014    2086    2276    2283    2314    2386    2598    2868    2907    3178
            3217    3495    3542    3818    3865    4141    4188    4465    4512    4753    4800    5040    5087    5326
            5373    5615    5662    5778    6052    6095    6372    6423    6698    6749    7026    7077    7354    7405
            7646    7697    7937    7988    8227    8278    8516    8567    8780    8793    8847    8868
.LCJP        948    1081    1093    1114    1343    1349    1377    1398    1408    1412    1414    1751    1753    1755
            1765    1789    1975    1988    2014    2086    2276    2283    2314    2386    2598    2868    2907    3178
            3217    3495    3542    3818    3865    4141    4188    4465    4512    4753    4800    5040    5087    5326
            5373    5615    5662    5778    6052    6095    6372    6423    6698    6749    7026    7077    7354    7405
            7646    7697    7937    7988    8227    8278    8516    8567    8780    8793    8847    8868
.LCJS       1330    1341    1749    1994    2289
.LCONT       273     280     287     296     305     314     323     332     341     350     359     368     377     386
             395     404     413     422     431     440     449     458     467     476     485     494     503     512
             521     530     539     548     557     566     575     584     593     611     613     615     740     747
             754     763     772     781     790     799     808     824     826     828     830     943     955     957
             959     964    1071    1076    1088    1100    1112    1118    1120    1332    1334    1336    1347    1356
            1358    1365    1375    1381    1383    1392    1402    1404    1406    1410    1421    1428    1430    1588
            1590    1592    1708    1710    1712    1738    1740    1742    1744    1759    1761    1763    1777    1787
            1793    1795    1979    1981    1984    1986    1992    2002    2012    2018    2020    2027    2029    2031
            2033    2035    2037    2039    2041    2043    2045    2047    2049    2051    2053    2055    2057    2059
            2061    2063    2090    2092    2094    2100    2272    2274    2280    2287    2302    2312    2318    2320
            2327    2329    2331    2333    2335    2337    2339    2341    2343    2345    2347    2349    2351    2353
            2355    2357    2359    2361    2363    2372    2390    2392    2394    2400    2596    2600    2857    2859
            2872    2874    2897    2899    2905    2911    3167    3169    3182    3184    3207    3209    3215    3221
            3484    3486    3499    3501    3510    3516    3532    3534    3540    3546    3807    3809    3822    3824
```

```
          3833  3839  3855  3857  3863  3869  4130  4132  4145  4147  4156  4162  4178  4180
          4186  4192  4454  4456  4469  4471  4480  4486  4502  4504  4510  4516  4742  4744
          4757  4759  4768  4774  4790  4792  4798  4804  5029  5031  5044  5046  5055  5061
          5077  5079  5085  5091  5315  5317  5330  5332  5341  5347  5363  5365  5371  5377
          5604  5606  5619  5621  5630  5636  5652  5654  5660  5666  5776  5780  6041  6043
          6056  6058  6060  6062  6085  6087  6093  6099  6361  6363  6376  6378  6380  6382
          6391  6397  6413  6415  6421  6427  6687  6689  6702  6704  6706  6708  6717  6723
          6739  6741  6747  6753  7015  7017  7030  7032  7034  7036  7045  7051  7067  7069
          7075  7081  7343  7345  7358  7360  7362  7364  7373  7379  7395  7397  7403  7409
          7635  7637  7650  7652  7654  7656  7665  7671  7687  7689  7695  7701  7926  7928
          7941  7943  7945  7947  7956  7962  7978  7980  7986  7992  8216  8218  8231  8233
          8235  8237  8246  8252  8268  8270  8276  8282  8505  8507  8520  8522  8524  8526
          8535  8541  8557  8559  8565  8571  8762  8764  8766  8768  8770  8772  8774  8776
          8778  8797  8799  8801  8803  8837  8839  8845  8851  8858  8860  8866  8872
.LCRTN    1416  1813  2065  2365
.LCRY      282   289   298   307   316   325   334   343   352   361   370   379   388   397
           406   415   424   433   442   451   460   469   478   487   496   505   514   523
           532   541   550   559   568   577   586   595   749   756   765   774   783   792
           801   810  1336  1347  1763  1986  1994  2274  2292  2899  3209  3534  3857  4180
          4504  4792  5079  5365  5654  6087  6415  6741  7069  7397  7689  7980  8270  8559
          8762  8780  8839  8860
.LD        273   275   280   282   287   289   296   298   305   307   314   316   323   325
           332   334   341   343   350   352   359   361   368   370   377   379   386   388
           395   397   404   406   413   415   422   424   431   433   440   442   449   451
           458   460   467   469   476   478   485   487   494   496   503   505   512   514
           521   523   530   532   539   541   548   550   557   559   566   568   575   577
           584   586   593   595   597   599   604   609   611   613   615   740   742   747
           749   754   756   763   765   772   774   781   783   790   792   799   801   808
           810   817   822   824   826   828   830   938   943   948   950   955   957   959
          1062  1064  1066  1071  1076  1081  1083  1088  1093  1095  1108  1110  1112  1118
          1120  1330  1332  1334  1336  1341  1343  1347  1349  1358  1371  1373  1375  1381
          1383  1404  1406  1408  1410  1430  1437  1444  1586  1588  1590  1592  1676  1678
          1680  1682  1684  1691  1693  1695  1697  1699  1706  1708  1710  1712  1714  1716
          1718  1720  1722  1724  1726  1728  1730  1732  1734  1736  1738  1740  1742  1744
          1761  1763  1765  1783  1785  1787  1793  1795  1802  1809  1811  1981  1984  1986
          1992  1994  2008  2010  2012  2018  2020  2072  2092  2094  2096  2098  2102  2269
          2272  2274  2280  2287  2289  2292  2308  2310  2312  2318  2320  2372  2392  2394
          2396  2398  2402  2485  2487  2594  2596  2598  2600  2855  2857  2859  2861  2868
          2870  2872  2874  2883  2897  2899  2903  2905  2907  2909  2911  2913  3165  3167
          3169  3171  3178  3180  3182  3184  3193  3207  3209  3213  3215  3217  3219  3221
          3223  3482  3484  3486  3488  3495  3497  3499  3501  3510  3512  3514  3516  3518
          3532  3534  3538  3540  3542  3544  3546  3548  3805  3807  3809  3811  3818  3820
          3822  3824  3833  3835  3837  3839  3841  3855  3857  3861  3863  3865  3867  3869
          3871  4128  4130  4132  4134  4141  4143  4145  4147  4156  4158  4160  4162  4164
          4178  4180  4184  4186  4188  4190  4192  4194  4452  4454  4456  4458  4465  4467
          4469  4471  4480  4482  4484  4486  4488  4502  4504  4508  4510  4512  4514  4516
          4518  4740  4742  4744  4746  4753  4755  4757  4759  4768  4770  4772  4774  4776
          4790  4792  4796  4798  4800  4802  4804  4806  5027  5029  5031  5033  5040  5042
          5044  5046  5055  5057  5059  5061  5063  5077  5079  5083  5085  5087  5089  5091
          5093  5313  5315  5317  5319  5326  5328  5330  5332  5341  5343  5345  5347  5349
          5363  5365  5369  5371  5373  5375  5377  5379  5602  5604  5606  5608  5615  5617
          5619  5621  5630  5632  5634  5636  5638  5652  5654  5658  5660  5662  5664  5666
          5668  5774  5776  5778  5780  6039  6041  6043  6045  6052  6054  6056  6058  6060
```

```
      6062  6071  6085  6087  6091  6093  6095  6097  6099  6101  6359  6361  6363  6365      SEQ 1849
      6372  6374  6376  6378  6380  6382  6391  6393  6395  6397  6399  6413  6415  6419
      6421  6423  6425  6427  6429  6685  6687  6689  6691  6698  6700  6702  6704  6706
      6708  6717  6719  6721  6723  6725  6739  6741  6745  6747  6749  6751  6753  6755
      7013  7015  7017  7019  7026  7028  7030  7032  7034  7036  7045  7047  7049  7051
      7053  7067  7069  7073  7075  7077  7079  7081  7083  7341  7343  7345  7347  7354
      7356  7358  7360  7362  7364  7373  7375  7377  7379  7381  7395  7397  7401  7403
      7405  7407  7409  7411  7633  7635  7637  7639  7646  7648  7650  7652  7654  7656
      7665  7667  7669  7671  7673  7687  7689  7693  7695  7697  7699  7701  7703  7924
      7926  7928  7930  7937  7939  7941  7943  7945  7947  7956  7958  7960  7962  7964
      7978  7980  7984  7986  7988  7990  7992  7994  8214  8216  8218  8220  8227  8229
      8231  8233  8235  8237  8246  8248  8250  8252  8254  8268  8270  8274  8276  8278
      8280  8282  8284  8503  8505  8507  8509  8516  9518  8520  8522  8524  8526  8535
      8537  8539  8541  8543  8557  8559  8563  8565  8567  8569  8571  8573  8758  8762
      8778  8780  8782  8786  8793  8795  8797  8799  8801  8803  8805  8812  8819  8823
      8837  8839  8843  8845  8847  8849  8851  8853  8858  8860  8864  8866  8868  8870
      8872  8874
.LJ    194   196   198   200   202   204   206   208   210   212   214   216   218   220
       222   224   226   228   230   232   234   236   238   240   242   244   246   248
       250   252   254   256   258   260   262   264   266   268   275   282   289   291
       298   300   307   309   316   318   325   327   334   336   343   345   352   354
       361   363   370   372   379   [illegible]1   388   390   397   399   406   408   415   417
       424   426   433   435   442   [illegible]4   451   453   460   462   469   471   478   480
       487   489   496   498   505   507   514   516   523   525   532   534   541   543
       550   552   559   561   568   570   577   579   586   588   595   597   599   604
       609   617   717   719   721   723   725   727   729   731   733   735   742   749
       756   758   765   767   774   776   783   785   792   794   801   803   810   812
       817   822   832   938   948   950   966   971   973  1062  1064  1066  1081  1083
      1093  1095  1102  1108  1110  1114  1116  1122  1127  1129  1330  1341  1343  1345
      1349  1351  1360  1371  1373  1377  1379  1385  1394  1396  1398  1400  1408  1412
      1414  1423  1432  1437  1439  1444  1446  1456  1458  1586  1676  1678  1680  1682
      1684  1691  1693  1695  1697  1699  1706  1714  1716  1718  1720  1722  1724  1726
      1728  1730  1732  1734  1736  1749  1751  1753  1755  1757  1765  1767  1773  1775
      1783  1785  1789  1791  1797  1802  1804  1809  1811  1823  1825  1975  1988  1990
      1994  1997  2008  2010  2014  2016  2022  2070  2072  2080  2082  2084  2086  2088
      2096  2098  2102  2107  2109  2269  2276  2278  2283  2285  2289  2292  2298  2300
      2308  2310  2314  2316  2322  2370  2380  2382  2384  2386  2388  2396  2398  2402
      2407  2409  2485  2487  2594  2598  2602  285[illegible]  2861  2866  2868  2870  2881  2890
      2901  2903  2907  2909  2913  3165  3171  3176  3178  3180  3191  3200  3211  3213
      3217  3219  3223  3482  3488  3493  3495  3497  3508  3512  3514  3525  3536  3538
      3542  3544  3548  3805  3811  3816  3818  3820  3831  3835  3837  3848  3859  3861
      3865  3867  3871  4128  4134  4139  4141  4143  4154  4158  4160  4171  4182  4184
      4188  4190  4194  4452  4458  4463  4465  4467  4478  4482  4484  4495  4506  4508
      4512  4514  4518  4740  4746  4751  4753  4755  4766  4770  4772  4783  4794  4796
      4800  4802  4806  5027  5033  5038  5040  5042  5053  5057  5059  5070  5081  5083
      5087  5089  5093  5313  5319  5324  5326  5328  5339  5343  5345  5356  5367  5369
      5373  5375  5379  5602  5608  5613  5615  5617  5628  5632  5634  5645  5656  5658
      5662  5664  5668  5774  5778  5782  6039  6045  6050  6052  6054  6069  6078  6089
      6091  6095  6097  6101  6359  6365  6370  6372  6374  6389  6393  6395  6406  6417
      6419  6423  6425  6429  6685  6691  6696  6698  6700  6715  6719  6721  6732  6743
      6745  6749  6751  6755  7013  7019  7024  7026  7028  7043  7047  7049  7060  7071
      7073  7077  7079  7083  7341  7347  7352  7354  7356  7371  7375  7377  7388  7399
      7401  7405  7407  7411  7633  7639  7644  7646  7648  7663  7667  7669  7680  7691
```

```
            7693    7697    7699    7703    7924    7930    7935    7937    7939    7954    7958    7960    7971    7982        SEQ 1850
            7984    7988    7990    7994    8214    8220    8225    8227    8229    8244    8248    8250    8261    8272
            8274    8278    8280    8284    8503    8509    8514    8516    8518    8533    8537    8539    8550    8561
            8563    8567    8569    8573    8758    8760    8780    8782    8784    8786    8791    8793    8795    8810
            8819    8821    8823    8825    8830    8841    8843    8847    8849    8853    8862    8864    8868    8870
            8874
.LJMAP       194     196     198     200     202     204     206     208     210     212     214     216     218     220
             222     224     226     228     230     232     234     236     238     240     242     244     246     248
             250     252     254     256     258     260     262     264     266     268     275     282     291     300
             309     318     327     336     345     354     363     372     381     390     399     408     417     426
             435     444     453     462     471     480     489     498     507     516     525     534     543     552
             561     570     579     588     599     604     617     717     719     721     723     725     727     729
             731     733     735     742     749     758     767     776     785     794     803     812     817     832
             938     950     966     971     973    1062    1064    1066    1082    1095    1102    1116    1122    1127
            1129    1345    1351    1360    1379    1385    1400    1423    1432    1437    1439    1444    1446    1456
            1458    1586    1676    1691    1706    1757    1767    1791    1797    1802    1804    1823    1825    1990
            2016    2022    2070    2080    2082    2084    2088    2102    2107    2109    2278    2285    2316    2322
            2370    2380    2382    2384    2388    2402    2407    2409    2485    2487    2594    2602    2855    2861
            2866    2870    2881    2890    2909    2913    3165    3171    3176    3180    3191    3200    3219    3223
            3482    3488    3493    3497    3508    3525    3544    3548    3805    3811    3816    3820    3831    3848
            3867    3871    4128    4134    4139    4143    4154    4171    4190    4194    4452    4458    4463    4467
            4478    4495    4514    4518    4740    4746    4751    4755    4766    4783    4802    4806    5027    5033
            5038    5042    5053    5070    5089    5093    5313    5319    5324    5328    5339    5356    5375    5379
            5602    5608    5613    5617    5628    5645    5664    5668    5774    5782    6039    6045    6050    6054
            6069    6078    6097    6101    6359    6365    6370    6374    6389    6406    6425    6429    6685    6691
            6696    6700    6715    6732    6751    6755    7013    7019    7024    7028    7043    7060    7079    7083
            7341    7347    7352    7356    7371    7388    7407    7411    7633    7639    7644    7648    7663    7680
            7699    7703    7924    7930    7935    7939    7954    7971    7990    7994    8214    8220    8225    8229
            8244    8261    8280    8284    8503    8509    8514    8518    8533    8550    8569    8573    8758    8782
            8786    8791    8795    8810    8819    8821    8823    8825    8830    8849    8853    8870    8874
.LJZ        1451    1686    1701    1818    1977    2074    2374    2876    2883    2885    2892    3186    3193    3195
            3202    3503    3518    3520    3527    3826    3841    3843    3850    4149    4164    4166    4173    4473
            4488    4490    4497    4761    4776    4778    4785    5048    5063    5065    5072    5334    5349    5351
            5358    5623    5638    5640    5647    6064    6071    6073    6080    6384    6399    6401    6408    6710
            6725    6727    6734    7038    7053    7055    7062    7366    7381    7383    7390    7658    7673    7675
            7682    7949    7964    7966    7973    8239    8254    8256    8263    8528    8543    8545    8552    8805
            8812    8814    8832
.LLDCT       289     298     307     316     325     334     343     352     361     370     379     388     397     406
             415     424     433     442     451     460     469     478     487     496     505     514     523     532
             541     550     559     568     577     586     595     756     765     774     783     792     801     810
            1108    1371    1394    1678    1682    1693    1697    1714    1718    1722    1726    1730    1734    1773
            1783    1809    2008    2072    2096    2269    2298    2308    2396    2901    3211    3512    3536    3835
            3859    4158    4182    4482    4506    4770    4794    5057    5081    5343    5367    5632    5656    6089
            6393    6417    6719    6743    7047    7071    7375    7399    7667    7691    7958    7982    8248    8272
            8537    8561    8760    8841    8862
.LMGC        611     613     615     824     826     828     830     943     955     957     959     964     971     973
            1071    1076    1088    1100    1108    1112    1114    1118    1120    1127    1129    1356    1358    1365
            1371    1375    1377    1381    1383    1392    1404    1406    1421    1428    1430    1437    1444    1451
            1456    1458    1586    1588    1590    1592    1678    1693    1706    1708    1710    1712    1749    1761
            1763    1777    1783    1787    1789    1793    1795    1802    1818    1823    1825    1979    1981    1984
            1986    1992    1994    2002    2008    2012    2014    2018    2020    2070    2072    2074    2080    2082
            2084    2090    2092    2096    2100    2107    2109    2272    2274    2280    2287    2289    2302    2308
            2312    2314    2318    2320    2370    2372    2374    2380    2382    2384    2390    2392    2396    2400
            2407  2409  2485  2487  2594  2596  2600  2857  2859  2872  2874  2881  2883  2890     SEQ 1851
            2892  2905  2907  2911  2913  3167  3169  3182  3184  3191  3193  3200  3202  3215
            3217  3221  3223  3484  3486  3499  3501  3508  3510  3512  3516  3518  3525  3527
            3540  3542  3546  3548  3807  3809  3822  3824  3831  3833  3835  3839  3841  3848
            3850  3863  3865  3869  3871  4130  4132  4145  4147  4154  4156  4158  4162  4164
            4171  4173  4186  4188  4192  4194  4454  4456  4469  4471  4478  4480  4482  4486
            4488  4495  4497  4510  4512  4516  4518  4742  4744  4757  4759  4766  4768  4770
            4774  4776  4783  4785  4798  4800  4804  4806  5029  5031  5044  5046  5053  5055
            5057  5061  5063  5070  5072  5085  5087  5091  5093  5315  5317  5330  5332  5339
            5341  5343  5347  5349  5356  5358  5371  5373  5377  5379  5604  5606  5619  5621
            5628  5630  5632  5636  5638  5645  5647  5660  5662  5666  5668  5774  5776  5780
            6041  6043  6056  6058  6060  6062  6069  6071  6078  6080  6093  6095  6099  6101
            6361  6363  6376  6378  6380  6382  6389  6391  6393  6397  6399  6406  6408  6421
            6423  6427  6429  6687  6689  6702  6704  6706  6708  6715  6717  6719  6723  6725
            6732  6734  6747  6749  6753  6755  7015  7017  7030  7032  7034  7036  7043  7045
            7047  7051  7053  7060  7062  7075  7077  7081  7083  7343  7345  7358  7360  7362
            7364  7371  7373  7375  7379  7381  7388  7390  7403  7405  7409  7411  7635  7637
            7650  7652  7654  7656  7663  7665  7667  7671  7673  7680  7682  7695  7697  7701
            7703  7926  7928  7941  7943  7945  7947  7954  7956  7958  7962  7964  7971  7973
            7986  7988  7992  7994  8216  8218  8231  8233  8235  8237  8244  8246  8248  8252
            8254  8261  8263  8276  8278  8282  8284  8505  8507  8520  8522  8524  8526  8533
            8535  8537  8541  8543  8550  8552  8565  8567  8571  8573  8762  8764  8766  8768
            8770  8772  8774  8776  8778  8797  8799  8801  8803  8810  8812  8819  8823  8830
            8832  8845  8847  8851  8853  8866  8868  8872  8874
.LOENA       611   615   824   830   959  1071  1112  1120  1358  1375  1383  1430  1437  1444
            1761  1787  1795  1802  1984  1992  2012  2020  2272  2287  2312  2320  2487  2600
            2905  2913  3215  3223  3516  3540  3548  3839  3863  3871  4162  4186  4194  4486
            4510  4518  4774  4798  4806  5061  5085  5093  5347  5371  5379  5636  5660  5668
            5780  6093  6101  6397  6421  6429  6723  6747  6755  7051  7075  7083  7379  7403
            7411  7671  7695  7703  7962  7986  7994  8252  8276  8284  8541  8565  8573  8819
            8823  8845  8853  8866  8874
.LOR         824   828   957   959  1108  1110  1112  1118  1120  1332  1349  1358  1371  1373
            1375  1381  1383  1406  1437  1444  1586  1588  1590  1592  1678  1680  1684  1693
            1695  1699  1706  1708  1710  1712  1716  1720  1724  1728  1732  1736  1738  1740
            1742  1744  1783  1785  1787  1793  1795  1802  1981  1984  1992  2008  2010  2012
            2018  2020  2072  2092  2094  2096  2098  2272  2280  2287  2308  2310  2312  2318
            2320  2372  2392  2394  2396  2398  2485  2487  2596  2598  2600  2859  2874  2905
            2911  2913  3169  3184  3215  3221  3223  3486  3501  3512  3514  3516  3540  3546
            3548  3809  3824  3835  3837  3839  3863  3869  3871  4132  4147  4158  4160  4162
            4186  4192  4194  4456  4471  4482  4484  4486  4510  4516  4518  4744  4759  4770
            4772  4774  4798  4804  4806  5031  5046  5057  5059  5061  5085  5091  5093  5317
            5332  5343  5345  5347  5371  5377  5379  5606  5621  5632  5634  5636  5660  5666
            5668  5776  5778  5780  6043  6062  6093  6099  6101  6363  6382  6393  6395  6397
            6421  6427  6429  6689  6708  6719  6721  6723  6747  6753  6755  7017  7036  7047
            7049  7051  7075  7081  7083  7345  7364  7375  7377  7379  7403  7409  7411  7637
            7656  7667  7669  7671  7695  7701  7703  7928  7947  7958  7960  7962  7986  7992
            7994  8218  8237  8248  8250  8252  8276  8282  8284  8507  8526  8537  8539  8541
            8565  8571  8573  8778  8803  8805  8819  8823  8845  8851  8853  8866  8872  8874
.LPLUS       282   289   298   307   316   325   334   343   352   361   370   379   388   397
             406   415   424   433   442   451   460   469   478   487   496   505   514   523
             532   541   550   559   568   577   586   595   597   609   749   756   765   774
             783   792   801   810   822  1336  1347  1763  1986  1994  2274  2292  2899  2903
            3209  3213  3534  3538  3857  3861  4180  4184  4504  4508  4792  4796  5079  5083
```

```
              5365    5369    5654    5658    6087    6091    6415    6419    6741    6745    7069    7073    7397    7401
              7689    7693    7980    7984    8270    8274    8559    8563    8762    8839    8843    8860    8864
.LRPCT         597     609     822    1110    1373    1396    1680    1684    1695    1699    1716    1720    1724    1728
              1732    1736    1775    1785    1811    1997    2010    2098    2292    2300    2310    2398    2903    3213
              3514    3538    3837    3861    4160    4184    4484    4508    4772    4796    5059    5083    5345    5369
              5634    5658    6091    6395    6419    6721    6745    7049    7073    7377    7401    7669    7693    7960
              7984    8250    8274    8539    8563    8784    8843    8864
.LSOA          611     615     824     830     959    1071    1112    1118    1120    1332    1334    1336    1358    1375
              1381    1383    1430    1437    1444    1738    1761    1787    1793    1795    1802    1992    2012    2018
              2020    2072    2094    2312    2318    2320    2372    2394    2487    2594    2598    2600    2905    2911
              2913    3215    3221    3223    3514    3516    3540    3546    3548    3837    3839    3863    3869    3871
              4160    4162    4186    4192    4194    4484    4486    4510    4516    4518    4772    4774    4798    4804
              4806    5059    5061    5085    5091    5093    5345    5347    5371    5377    5379    5634    5636    5660
              5666    5668    5774    5778    5780    6093    6099    6101    6395    6397    6421    6427    6429    6721
              6723    6747    6753    6755    7049    7051    7075    7081    7083    7377    7379    7403    7409    7411
              7669    7671    7695    7701    7703    7960    7962    7986    7992    7994    8250    8252    8276    8282
              8284    8539    8541    8565    8571    8573    8805    8819    8823    8845    8851    8853    8866    8872
              8874
.LSOB         1110    1347    1349    1373    1404    1408    1410    1680    1684    1695    1699    1716    1720    1724
              1728    1732    1736    1763    1765    1785    1986    1994    2010    2098    2102    2269    2274    2287
              2292    2310    2398    2402    2899    3209    3534    3857    4180    4504    4792    5079    5365    5654
              6087    6415    6741    7069    7397    7689    7980    8270    8559    8758    8762    8839    8860
.LSAB          275     282     289     298     307     316     325     334     343     352     361     370     379     388
               397     406     415     424     433     442     451     460     469     478     487     496     505     514
               523     532     541     550     559     568     577     586     595     597     599     609     742     749
               756     765     774     783     792     801     810     822    1066    1343    1740    1742    1744    1984
              2272    2903    3213    3538    3861    4184    4508    4796    5083    5369    5658    6091    6419    6745
              7073    7401    7693    7984    8274    8563    8780    8843    8864
.LSDO          273     280     287     296     305     314     323     332     341     350     359     368     377     386
               395     404     413     422     431     440     449     458     467     476     485     494     503     512
               521     530     539     548     557     566     575     584     593     604     613     740     747     754
               763     772     781     790     799     808     817     826     828     938     957    1062    1064    1108
              1371    1406    1586    1588    1590    1592    1676    1678    1691    1693    1706    1708    1710    1712
              1783    1981    2008    2092    2096    2280    2308    2392    2396    2485    2596    2859    2874    2897
              3169    3184    3207    3486    3501    3512    3518    3532    3809    3824    3835    3841    3855    4132
              4147    4158    4164    4178    4456    4471    4482    4488    4502    4744    4759    4770    4776    4790
              5031    5046    5057    5063    5077    5317    5332    5343    5349    5363    5606    5621    5632    5638
              5652    5776    6043    6062    6085    6363    6382    6393    6399    6413    6689    6708    6719    6725
              6739    7017    7036    7047    7053    7067    7345    7364    7375    7381    7395    7637    7656    7667
              7673    7687    7928    7947    7958    7964    7978    8218    8237    8248    8254    8268    8507    8526
              8537    8543    8557    8778    8803    8837    8858
.LSELC         943     955     964     971    1076    1088    1100    1127    1356    1365    1392    1404    1421    1428
              1456    1749    1763    1777    1823    1979    2002    2072    2074    2080    2084    2090    2100    2107
              2289    2302    2372    2374    2380    2384    2390    2400    2407    2857    2872    2883    2890    3182
              3193    3200    3499    3510    3525    3822    3833    3848    4145    4156    4171    4469    4480    4495
              4757    4768    4783    5044    5055    5070    5330    5341    5356    5619    5630    5645    6060    6071
              6078    6380    6391    6406    6706    6717    6732    7034    7045    7060    7362    7373    7388    7654
              7665    7680    7945    7956    7971    8235    8246    8261    8524    8535    8550    8801    8812    8830
.LSELE         615     830     959    1112    1114    1118    1120    1358    1375    1377    1381    1383    1430    1437
              1444    1787    1789    1793    1795    1802    1984    1986    1992    1994    2012    2014    2018    2020
              2272    2274    2280    2312    2314    2318    2320    2487    2600    2905    2907    2911    2913    3215
              3217    3221    3223    3540    3542    3546    3548    3863    3865    3869    3871    4186    4188    4192
              4194    4510    4512    4516    4518    4798    4800    4804    4806    5085    5087    5091    5093    5371
```

```
               5373      5377      5379      5660      5662      5666      5668      5780      6093      6095      6099      6101      6421      6423
               6427      6429      6747      6749      6753      6755      7075      7077      7081      7083      7403      7405      7409      7411
               7695      7697      7701      7703      7986      7988      7992      7994      8276      8278      8282      8284      8565      8567
               8571      8573      8819      8823      8845      8847      8851      8853      8866      8868      8872      8874
.LSELF          611      1071      1761      2287      2594      3167      3484      3807      4130      4454      4742      5029      5315      5604
               5774      6041      6056      6058      6361      6376      6378      6687      6702      6704      7015      7030      7032      7343
               7358      7360      7635      7650      7652      7926      7941      7943      8216      8231      8233      8505      8520      8522
               8762      8764      8766      8768      8770      8772      8774      8776      8797      8799
.LSELM          613       824       828       957       973      1129      1406      1451      1458      1818      1825      1981      2070      2082
               2092      2109      2370      2382      2392      2409      2485      2596      2859      2874      2881      2892      3169      3184
               3191      3202      3486      3501      3508      3516      3527      3809      3824      3831      3839      3850      4132      4147
               4154      4162      4173      4456      4471      4478      4486      4497      4744      4759      4766      4774      4785      5031
               5046      5053      5061      5072      5317      5332      5339      5347      5358      5606      5621      5628      5636      5647
               5776      6043      6062      6069      6080      6363      6382      6389      6397      6408      6689      6708      6715      6723
               6734      7017      7036      7043      7051      7062      7345      7364      7371      7379      7390      7637      7656      7663
               7671      7682      7928      7947      7954      7962      7973      8218      8237      8244      8252      8263      8507      8526
               8533      8541      8552      8778      8803      8810      8832
.LSELP          826      3518      3841      4164      4488      4776      5063      5349      5638      6399      6725      7053      7381      7673
               7964      8254      8543
.LSKCN         1108      1371      1586      1588      1590      1592      1678      1693      1706      1708      1710      1712      1783      2008
               2096      2308      2396      3512      3835      4158      4482      4770      5057      5343      5632      6393      6719      7047
               7375      7667      7958      8248      8537
.LSMIN         1765      8780
.LXNOR          275       742      1066      1408      1430
.LXOR          1343
.MA             275       282       289       298       307       316       325       334       343       352       361       370       379       388
                397       406       415       424       433       442       451       460       469       478       487       496       505       514
                523       532       541       550       559       568       577       586       595       597       599       609       611       615
                742       749       756       765       774       783       792       801       810       822       824       830      1112      1118
               1120      1332      1334      1336      1343      1358      1375      1381      1383      1430      1437      1444      1740      1742
               1744      1761      1787      1793      1795      1802      1984      1992      2012      2018      2020      2072      2094      2272
               2312      2318      2320      2372      2394      2903      2905      2911      2913      3213      3215      3221      3223      3538
               3540      3546      3548      3861      3863      3869      3871      4184      4186      4192      4194      4508      4510      4516
               4518      4796      4798      4804      4806      5083      5085      5091      5093      5369      5371      5377      5379      5658
               5660      5666      5668      6091      6093      6099      6101      6419      6421      6427      6429      6745      6747      6753
               6755      7073      7075      7081      7083      7401      7403      7409      7411      7693      7695      7701      7703      7984
               7986      7992      7994      8274      8276      8282      8284      8563      8565      8571      8573      8780      8805      8819
               8823      8843      8845      8851      8853      8864      8866      8872      8874
.MAND           273       280       287       296       305       314       323       332       341       350       359       368       377       386
                395       404       413       422       431       440       449       458       467       476       485       494       503       512
                521       530       539       548       557       566       575       584       593       604       740       747       754       763
                772       781       790       799       808       817       938      1062      1064      1334      1404      1410      1676      1691
               2102      2269      2402      2897      3207      3532      3855      4178      4502      4790      5077      5363      5652      6085
               6413      6739      7067      7395      7687      7978      8268      8557      8758      8837      8858
.MB             273       275       280       282       287       289       296       298       305       307       314       316       323       325
                332       334       341       343       350       352       359       361       368       370       377       379       386       388
                395       397       404       406       413       415       422       424       431       433       440       442       449       451
                458       460       467       469       476       478       485       487       494       496       503       505       512       514
                521       523       530       532       539       541       548       550       557       559       566       568       575       577
                584       586       593       595       597       599       604       609       613       740       742       747       749       754
                756       763       765       772       774       781       783       790       792       799       801       808       810       817
                822       828      1108      1110      1332      1334      1336      1343      1347      1349      1371      1373      1404      1406
               1408      1410      1588      1590      1592      1676      1678      1680      1684      1691      1693      1695      1699      1708
```

```
            1710    1712    1716    1720    1724    1728    1732    1736    1738    1740    1742    1744    1763    1765
            1783    1785    1981    1984    1986    1994    2008    2010    2072    2092    2094    2096    2098    2102
            2269    2272    2274    2280    2287    2292    2308    2310    2372    2392    2394    2396    2398    2402
            2859    2874    2897    2899    2903    3169    3184    3207    3209    3213    3486    3501    3532    3534
            3538    3809    3824    3855    3857    3861    4132    4147    4178    4180    4184    4456    4471    4502
            4504    4508    4744    4759    4790    4792    4796    5031    5046    5077    5079    5083    5317    5332
            5363    5365    5369    5606    5621    5652    5654    5658    6043    6062    6085    6087    6091    6363
            6382    6413    6415    6419    6689    6708    6739    6741    6745    7017    7036    7067    7069    7073
            7345    7364    7395    7397    7401    7637    7656    7687    7689    7693    7928    7947    7978    7980
            7984    8218    8237    8268    8270    8274    8507    8526    8557    8559    8563    8758    8762    8778
            8780    8803    8805    8837    8839    8843    8858    8860    8864
.MBAD       2602    5782
.MCCCA       948    1081    1093    1398    1755    1975    2086    2283    2386    2868    3178    3495    3818    4141
            4465    4753    5040    5326    5615    6052    6372    6698    7026    7354    7646    7937    8227    8516
            8793
.MCCCE      1412    1751
.MCCCP      1414    1753
.MCCER      1988    2276
.MCCFZ      1343    1765    2598    5778    8780
.MCCGC      1114    1377    1789    2014    2314    2907    3217    3542    3865    4188    4512    4800    5087    5373
            5662    6095    6423    6749    7077    7405    7697    7988    8278    8567    8847    8868
.MCCLW      1408
.MCCMB      1349
.MCENA       948    1081    1093    1114    1343    1349    1377    1398    1408    1412    1414    1751    1753    1755
            1765    1789    1975    1988    2014    2086    2276    2283    2314    2386    2598    2868    2907    3178
            3217    3495    3542    3818    3865    4141    4188    4465    4512    4753    4800    5040    5087    5326
            5373    5615    5662    5778    6052    6095    6372    6423    6698    6749    7026    7077    7354    7405
            7646    7697    7937    7988    8227    8278    8516    8567    8780    8793    8847    8868
.MCJP        948    1081    1093    1114    1343    1349    1377    1398    1408    1412    1414    1751    1753    1755
            1765    1789    1975    1988    2014    2086    2276    2283    2314    2386    2598    2868    2907    3178
            3217    3495    3542    3818    3865    4141    4188    4465    4512    4753    4800    5040    5087    5326
            5373    5615    5662    5778    6052    6095    6372    6423    6698    6749    7026    7077    7354    7405
            7646    7697    7937    7988    8227    8278    8516    8567    8780    8793    8847    8868
.MCJS       1330    1341    1749    1994    2289
.MCONT       273     280     287     296     305     314     323     332     341     350     359     368     377     386
             395     404     413     422     431     440     449     458     467     476     485     494     503     512
             521     530     539     548     557     566     575     584     593     611     613     615     740     747
             754     763     772     781     790     799     808     824     826     828     830     943     955     957
             959     964    1071    1076    1088    1100    1112    1118    1120    1332    1334    1336    1347    1356
            1358    1365    1375    1381    1383    1392    1402    1404    1406    1410    1421    1428    1430    1588
            1590    1592    1708    1710    1712    1738    1740    1742    1744    1759    1761    1763    1777    1787
            1793    1795    1979    1981    1984    1986    1992    2002    2012    2018    2020    2027    2029    2031
            2033    2035    2037    2039    2041    2043    2045    2047    2049    2051    2053    2055    2057    2059
            2061    2063    2090    2092    2094    2100    2272    2274    2280    2287    2302    2312    2318    2320
            2327    2329    2331    2333    2335    2337    2339    2341    2343    2345    2347    2349    2351    2353
            2355    2357    2359    2361    2363    2372    2390    2392    2394    2400    2596    2600    2857    2859
            2872    2874    2897    2899    2905    2911    3167    3169    3182    3184    3207    3209    3215    3221
            3484    3486    3499    3501    3510    3516    3532    3534    3540    3546    3807    3809    3822    3824
            3833    3839    3855    3857    3863    3869    4130    4132    4145    4147    4156    4162    4178    4180
            4186    4192    4454    4456    4469    4471    4480    4486    4502    4504    4510    4516    4742    4744
            4757    4759    4768    4774    4790    4792    4798    4804    5029    5031    5044    5046    5055    5061
            5077    5079    5085    5091    5315    5317    5330    5332    5341    5347    5363    5365    5371    5377
            5604    5606    5619    5621    5630    5636    5652    5654    5660    5666    5776    5780    6041    6043
```

```
           6056  6058  6060  6062  6085  6087  6093  6099  6361  6363  6376  6378  6380  6382
           6391  6397  6413  6415  6421  6427  6687  6689  6702  6704  6706  6708  6717  6723
           6739  6741  6747  6753  7015  7017  7030  7032  7034  7036  7045  7051  7067  7069
           7075  7081  7343  7345  7358  7360  7362  7364  7373  7379  7395  7397  7403  7409
           7635  7637  7650  7652  7654  7656  7665  7671  7687  7689  7695  7701  7926  7928
           7941  7943  7945  7947  7956  7962  7978  7980  7986  7992  8216  8218  8231  8233
           8235  8237  8246  8252  8268  8270  8276  8282  8505  8507  8520  8522  8524  8526
           8535  8541  8557  8559  8565  8571  8762  8764  8766  8768  8770  8772  8774  8776
           8778  8797  8799  8801  8803  8837  8839  8845  8851  8858  8860  8866  8872
.MCRTN     1416  1813  2065  2365
.MCRY       282   289   298   307   316   325   334   343   352   361   370   379   388   397
            406   415   424   433   442   451   460   469   478   487   496   505   514   523
            532   541   550   559   568   577   586   595   749   756   765   774   783   792
            801   810  1336  1347  1763  1986  1994  2274  2292  2899  3209  3534  3857  4180
           4504  4792  5079  5365  5654  6087  6415  6741  7069  7397  7689  7980  8270  8559
           8762  8780  8839  8860
.MD         273   275   280   282   287   289   296   298   305   307   314   316   323   325
            332   334   341   343   350   352   359   361   368   370   377   379   386   388
            395   397   404   406   413   415   422   424   431   433   440   442   449   451
            458   460   467   469   476   478   485   487   494   496   503   505   512   514
            521   523   530   532   539   541   548   550   557   559   566   568   575   577
            584   586   593   595   597   599   604   609   611   613   615   740   742   747
            749   754   756   763   765   772   774   781   783   790   792   799   801   808
            810   817   822   824   826   828   830   938   943   948   950   955   957   959
           1062  1064  1066  1071  1076  1081  1083  1088  1093  1095  1108  1110  1112  1118
           1120  1330  1332  1334  1336  1341  1343  1347  1349  1358  1371  1373  1375  1381
           1383  1404  1406  1408  1410  1430  1437  1444  1586  1588  1590  1592  1676  1678
           1680  1682  1684  1691  1693  1695  1697  1699  1706  1708  1710  1712  1714  1716
           1718  1720  1722  1724  1726  1728  1730  1732  1734  1736  1738  1740  1742  1744
           1761  1763  1765  1783  1785  1787  1793  1795  1802  1809  1811  1981  1984  1986
           1992  1994  2008  2010  2012  2018  2020  2072  2092  2094  2096  2098  2102  2269
           2272  2274  2280  2287  2289  2292  2308  2310  2312  2318  2320  2372  2392  2394
           2396  2398  2402  2485  2487  2594  2596  2598  2600  2855  2857  2859  2861  2868
           2870  2872  2874  2883  2897  2899  2903  2905  2907  2909  2911  2913  3165  3167
           3169  3171  3178  3180  3182  3184  3193  3207  3209  3213  3215  3217  3219  3221
           3223  3482  3484  3486  3488  3495  3497  3499  3501  3510  3512  3514  3516  3518
           3532  3534  3538  3540  3542  3544  3546  3548  3805  3807  3809  3811  3818  3820
           3822  3824  3833  3835  3837  3839  3841  3855  3857  3861  3863  3865  3867  3869
           3871  4128  4130  4132  4134  4141  4143  4145  4147  4156  4158  4160  4162  4164
           4178  4180  4184  4186  4188  4190  4192  4194  4452  4454  4456  4458  4465  4467
           4469  4471  4480  4482  4484  4486  4488  4502  4504  4508  4510  4512  4514  4516
           4518  4740  4742  4744  4746  4753  4755  4757  4759  4768  4770  4772  4774  4776
           4790  4792  4796  4798  4800  4802  4804  4806  5027  5029  5031  5033  5040  5042
           5044  5046  5055  5057  5059  5061  5063  5077  5079  5083  5085  5087  5089  5091
           5093  5313  5315  5317  5319  5326  5328  5330  5332  5341  5343  5345  5347  5349
           5363  5365  5369  5371  5373  5375  5377  5379  5602  5604  5606  5608  5615  5617
           5619  5621  5630  5632  5634  5636  5638  5652  5654  5658  5660  5662  5664  5666
           5668  5774  5776  5778  5780  6039  6041  6043  6045  6052  6054  6056  6058  6060
           6062  6071  6085  6087  6091  6093  6095  6097  6099  6101  6359  6361  6363  6365
           6372  6374  6376  6378  6380  6382  6391  6393  6395  6397  6399  6413  6415  6419
           6421  6423  6425  6427  6429  6685  6687  6689  6691  6698  6700  6702  6704  6706
           6708  6717  6719  6721  6723  6725  6739  6741  6745  6747  6749  6751  6753  6755
           7013  7015  7017  7019  7026  7028  7030  7032  7034  7036  7045  7047  7049  7051
```

```
          7053   7067   7069   7073   7075   7077   7079   7081   7083   7341   7343   7345   7347   7354
Q 1843    7356   7358   7360   7362   7364   7373   7375   7377   7379   7381   7395   7397   7401   7403    SEQ 1856
          7405   7407   7409   7411   7633   7635   7637   7639   7646   7648   7650   7652   7654   7656
          7665   7667   7669   7671   7673   7687   7689   7693   7695   7697   7699   7701   7703   7924
          7926   7928   7930   7937   7939   7941   7943   7945   7947   7956   7958   7960   7962   7964
          7978   7980   7984   7986   7988   7990   7992   7994   8214   8216   8218   8220   8227   8229
          8231   8233   8235   8237   8246   8248   8250   8252   8254   8268   8270   8274   8276   8278
          8280   8282   8284   8503   8505   8507   8509   8516   8518   8520   8522   8524   8526   8535
          8537   8539   8541   8543   8557   8559   8563   8565   8567   8569   8571   8573   8758   8762
          8778   8780   8782   8786   8793   8795   8797   8799   8801   8803   8805   8812   8819   8823
          8837   8839   8843   8845   8847   8849   8851   8853   8858   8860   8864   8866   8868   8870
          8872   8874
.MJ        194    196    198    200    202    204    206    208    210    212    214    216    218    220
           222    224    226    228    230    232    234    236    238    240    242    244    246    248
           250    252    254    256    258    260    262    264    266    268    275    282    289    291
           298    300    307    309    316    318    325    327    334    336    343    345    352    354
           361    363    370    372    379    381    388    390    397    399    406    408    415    417
           424    426    433    435    442    444    451    453    460    462    469    471    478    480
           487    489    496    498    505    507    514    516    523    525    532    534    541    543
           550    552    559    561    568    570    577    579    586    588    595    597    599    604
           609    617    717    719    721    723    725    727    729    731    733    735    742    749
           756    758    765    767    774    776    783    785    792    794    801    803    810    812
           817    822    832    938    948    950    966    971    973   1062   1064   1066   1081   1083
          1093   1095   1102   1108   1110   1114   1116   1122   1127   1129   1330   1341   1343   1345
          1349   1351   1360   1371   1373   1377   1379   1385   1394   1396   1398   1400   1408   1412
          1414   1423   1432   1437   1439   1444   1446   1456   1458   1586   1676   1678   1680   1682
          1684   1691   1693   1695   1697   1699   1706   1714   1716   1718   1720   1722   1724   1726
          1728   1730   1732   1734   1736   1749   1751   1753   1755   1757   1765   1767   1773   1775    CO
          1783   1785   1789   1791   1797   1802   1804   1809   1811   1823   1825   1975   1988   1990
          1994   1997   2008   2010   2014   2016   2022   2070   2072   2080   2082   2084   2086   2088
          2096   2098   2102   2107   2109   2269   2276   2278   2283   2285   2289   2292   2298   2300
          2308   2310   2314   2316   2322   2370   2380   2382   2384   2386   2388   2396   2398   2402
          2407   2409   2485   2487   2594   2598   2602   2855   2861   2866   2868   2870   2881   2890
          2901   2903   2907   2909   2913   3165   3171   3176   3178   3180   3191   3200   3211   3213
          3217   3219   3223   3482   3488   3493   3495   3497   3508   3512   3514   3525   3536   3538
          3542   3544   3548   3805   3811   3816   3818   3820   3831   3835   3837   3848   3859   3861
          3865   3867   3871   4128   4134   4139   4141   4143   4154   4158   4160   4171   4182   4184
          4188   4190   5194   4452   4458   4463   4465   4467   4478   4482   4484   4495   4506   4508
          4512   4514   4518   4740   4746   4751   4753   4755   4766   4770   4772   4783   4794   4796
          4800   4802   4806   5027   5033   5038   5040   5042   5053   5057   5059   5070   5081   5083
          5087   5089   5093   5313   5319   5324   5326   5328   5339   5343   5345   5356   5367   5369
          5373   5375   5379   5602   5608   5613   5615   5617   5628   5632   5634   5645   5656   5658
          5662   5664   5668   5774   5778   5782   6039   6045   6050   6052   6054   6069   6078   6089
          6091   6095   6097   6101   6359   6365   6370   6372   6374   6389   6393   6395   6406   6417
          6419   6423   6425   6429   6685   6691   6696   6698   6700   6715   6719   6721   6732   6743
          6745   6749   6751   6755   7013   7019   7024   7026   7028   7043   7047   7049   7060   7071
          7073   7077   7079   7083   7341   7347   7352   7354   7356   7371   7375   7377   7388   7399
          7401   7405   7407   7411   7633   7639   7644   7646   7648   7663   7667   7669   7680   7691
          7693   7697   7699   7703   7924   7930   7935   7937   7939   7954   7958   7960   7971   7982
          7984   7988   7990   7994   8214   8220   8225   8227   8229   8244   8248   8250   8261   8272
          8274   8278   8280   8284   8503   8509   8514   8516   8518   8533   8537   8539   8550   8561
          8563   8567   8569   8573   8758   8760   8780   8782   8784   8786   8791   8793   8795   8810
          8819   8821   8823   8825   8830   8841   8843   8847   8849   8853   8862   8864   8868   8870
```

```
                  8874
Q 1844  .MJMAP     194   196   198   200   202   204   206   208   210   212   214   216   218   220   SEQ 1857
                   222   224   226   228   230   232   234   236   238   240   242   244   246   248
                   250   252   254   256   258   260   262   264   266   268   275   282   291   300
                   309   318   327   336   345   354   363   372   381   390   399   408   417   426
                   435   444   453   462   471   480   489   498   507   516   525   534   543   552
                   561   570   579   588   599   604   617   717   719   721   723   725   727   729
                   731   733   735   742   749   758   767   776   785   794   803   812   817   832
                   938   950   966   971   973  1062  1064  1066  1083  1095  1102  1116  1122  1127
                  1129  1345  1351  1360  1379  1385  1400  1423  1432  1437  1439  1444  1446  1456
                  1458  1586  1676  1691  1706  1757  1767  1791  1797  1802  1804  1823  1825  1990
                  2016  2022  2070  2080  2082  2084  2088  2102  2107  2109  2278  2285  2316  2322
                  2370  2380  2382  2384  2388  2402  2407  2409  2485  2487  2594  2602  2855  2861
                  2866  2870  2881  2890  2909  2913  3165  3171  3176  3180  3191  3200  3219  3223
                  3482  3488  3493  3497  3508  3525  3544  3548  3805  3811  3816  3820  3831  3848
                  3867  3871  4128  4134  4139  4143  4154  4171  4190  4194  4452  4458  4463  4467
                  4478  4495  4514  4518  4740  4746  4751  4755  4766  4783  4802  4806  5027  5033
                  5038  5042  5053  5070  5089  5093  5313  5319  5324  5328  5339  5356  5375  5379
                  5602  5608  5613  5617  5628  5645  5664  5668  5774  5782  6039  6045  6050  6054
                  6069  6078  6097  6101  6359  6365  6370  6374  6389  6406  6425  6429  6685  6691
                  6696  6700  6715  6732  6751  6755  7013  7019  7024  7028  7043  7060  7079  7083
                  7341  7347  7352  7356  7371  7388  7407  7411  7633  7639  7644  7648  7663  7680
                  7699  7703  7924  7930  7935  7939  7954  7971  7990  7994  8214  8220  8225  8229
                  8244  8261  8280  8284  8503  8509  8514  8518  8533  8550  8569  8573  8758  8782
                  8786  8791  8795  8810  8819  8821  8823  8825  8830  8849  8853  8870  8874
        .MJZ      1451  1686  1701  1818  1977  2074  2374  2876  2883  2885  2892  3186  3193  3195
                  3202  3503  3518  3520  3527  3826  3841  3843  3850  4149  4164  4166  4173  4473
                  4488  4490  4497  4761  4776  4778  4785  5048  5063  5065  5072  5334  5349  5351
                  5358  5623  5638  5640  5647  6064  6071  6073  6080  6384  6399  6401  6408  6710
                  6725  6727  6734  7038  7053  7055  7062  7366  7381  7383  7390  7658  7673  7675
                  7682  7949  7964  7966  7973  8239  8254  8256  8263  8528  8543  8545  8552  8805
                  8812  8814  8832
        .MLDCT     289   298   307   316   325   334   343   352   361   370   379   388   397   406
                   415   424   433   442   451   460   469   478   487   496   505   514   523   532
                   541   550   559   568   577   586   595   756   765   774   783   792   801   810
                  1108  1371  1394  1678  1682  1693  1697  1714  1718  1722  1726  1730  1734  1773
                  1783  1809  2008  2072  2096  2269  2298  2308  2396  2901  3211  3512  3536  3835
                  3859  4158  4182  4482  4506  4770  4794  5057  5081  5343  5367  5632  5656  6089
                  6393  6417  6719  6743  7047  7071  7375  7399  7667  7691  7958  7982  8248  8272
                  8537  8561  8760  8841  8862
        .MMGC      611   613   615   824   826   828   830   943   955   957   959   964   971   973
                  1071  1076  1088  1100  1108  1112  1114  1118  1120  1127  1129  1356  1358  1365
                  1371  1375  1377  1381  1383  1392  1404  1406  1421  1428  1430  1437  1444  1451
                  1456  1458  1586  1588  1590  1592  1678  1693  1706  1708  1710  1712  1749  1761
                  1763  1777  1783  1787  1789  1793  1795  1802  1818  1823  1825  1979  1981  1984
                  1986  1992  1994  2002  2008  2012  2014  2018  2020  2070  2072  2074  2080  2082
                  2084  2090  2092  2096  2100  2107  2109  2272  2274  2280  2287  2289  2302  2308
                  2312  2314  2318  2320  2370  2372  2374  2380  2382  2384  2390  2392  2396  2400
                  2407  2409  2485  2487  2594  2596  2600  2857  2859  2872  2874  2881  2883  2890
                  2892  2905  2907  2911  2913  3167  3169  3182  3184  3191  3193  3200  3202  3215
                  3217  3221  3223  3484  3486  3499  3501  3508  3510  3512  3516  3519  3525  3527            CT
                  3540  3542  3546  3548  3807  3809  3822  3824  3831  3833  3835  3839  3841  3848
                  3850  3863  3865  3869  3871  4130  4132  4145  4147  4154  4156  4158  4162  4164
```

```
            4171    4173    4186    4188    4192    4194    4454    4456    4469    4471    4478    4480    4482    4486
            4488    4495    4497    4510    4512    4516    4518    4742    4744    4757    4759    4766    4768    4770
            4774    4776    4783    4785    4798    4800    4804    4806    5029    5031    5044    5046    5053    5055
            5057    5061    5063    5070    5072    5085    5087    5091    5093    5315    5317    5330    5332    5339
            5341    5343    5347    5349    5356    5358    5371    5373    5377    5379    5604    5606    5619    5621
            5628    5630    5632    5636    5638    5645    5647    5660    5662    5666    5668    5774    5776    5780
            6041    6043    6056    6058    6060    6062    6069    6071    6078    6080    6093    6095    6099    6101
            6361    6363    6376    6378    6380    6382    6389    6391    6393    6397    6399    6406    6408    6421
            6423    6427    6429    6687    6689    6702    6704    6706    6708    6715    6717    6719    6723    6725
            6732    6734    6747    6749    6753    6755    7015    7017    7030    7032    7034    7036    7043    7045
            7047    7051    7053    7060    7062    7075    7077    7081    7083    7343    7345    7358    7360    7362
            7364    7371    7373    7375    7379    7381    7388    7390    7403    7405    7409    7411    7635    7637
            7650    7652    7654    7656    7663    7665    7667    7671    7673    7680    7682    7695    7697    7701
            7703    7926    7928    7941    7943    7945    7947    7954    7956    7958    7962    7964    7971    7973
            7986    7988    7992    7994    8216    8218    8231    8233    8235    8237    8244    8246    8248    8252
            8254    8261    8263    8276    8278    8282    8284    8505    8507    8520    8522    8524    8526    8533
            8535    8537    8541    8543    8550    8552    8565    8567    8571    8573    8762    8764    8766    8768
            8770    8772    8774    8776    8778    8797    8799    8801    8803    8810    8812    8819    8823    8830
            8832    8845    8847    8851    8853    8866    8868    8872    8874
.MOENA       611     615     824     830     959    1071    1112    1120    1358    1375    1383    1430    1437    1444
            1761    1787    1795    1802    1984    1992    2012    2020    2272    2287    2312    2320    2487    2600
            2905    2913    3215    3223    3516    3540    3548    3839    3863    3871    4162    4186    4194    4486
            4510    4518    4774    4798    4806    5061    5085    5093    5347    5371    5379    5636    5660    5668
            5780    6093    6101    6397    6421    6429    6723    6747    6755    7051    7075    7083    7379    7403
            7411    7671    7695    7703    7962    7986    7994    8252    8276    8284    8541    8565    8573    8819
            8823    8845    8853    8866    8874
.MOR         824     828     957     959    1108    1110    1112    1118    1120    1332    1349    1358    1371    1373
            1375    1381    1383    1406    1437    1444    1586    1588    1590    1592    1678    1680    1684    1693
            1695    1699    1706    1708    1710    1712    1716    1720    1724    1728    1732    1736    1738    1740
            1742    1744    1783    1785    1787    1793    1795    1802    1981    1984    1992    2008    2010    2012
            2018    2020    2072    2092    2094    2096    2098    2272    2280    2287    2308    2310    2312    2318
            2320    2372    2392    2394    2396    2398    2485    2487    2596    2598    2600    2859    2874    2905
            2911    2913    3169    3184    3215    3221    3223    3486    3501    3512    3514    3516    3540    3546
            3548    3809    3824    3835    3837    3839    3863    3869    3871    4132    4147    4158    4160    4162
            4186    4192    4194    4456    4471    4482    4484    4486    4510    4516    4518    4744    4759    4770
            4772    4774    4798    4804    4806    5031    5046    5057    5059    5061    5085    5091    5093    5317
            5332    5343    5345    5347    5371    5377    5379    5606    5621    5632    5634    5636    5660    5666
            5668    5776    5778    5780    6043    6062    6093    6099    6101    6363    6382    6393    6395    6397
            6421    6427    6429    6689    6708    6719    6721    6723    6747    6753    6755    7017    7036    7047
            7049    7051    7075    7081    7083    7345    7364    7375    7377    7379    7403    7409    7411    7637
            7656    7667    7669    7671    7695    7701    7703    7928    7947    7958    7960    7962    7986    7992
            7994    8218    8237    8248    8250    8252    8276    8282    8284    8507    8526    8537    8539    8541
            8565    8571    8573    8778    8803    8805    8819    8823    8845    8851    8853    8866    8872    8874
.MPLUS       282     289     298     307     316     325     334     343     352     361     370     379     388     397
             406     415     424     433     442     451     460     469     478     487     496     505     514     523
             532     541     550     559     568     577     586     595     597     609     749     756     765     774
             783     792     801     810     822    1336    1347    1763    1986    1994    2274    2292    2899    2903
            3209    3213    3534    3538    3857    3861    4180    4184    4504    4508    4792    4796    5079    5083
            5365    5369    5654    5658    6087    6091    6415    6419    6741    6745    7069    7073    7397    7401
            7689    7693    7980    7984    8270    8274    8559    8563    8762    8839    8843    8860    8864
.MRPCT       597     609     822    1110    1373    1396    1680    1684    1695    1699    1716    1720    1724    1728
            1732    1736    1775    1785    1811    1997    2010    2098    2292    2300    2310    2398    2903    3213
            3514    3538    3837    3861    4160    4184    4484    4508    4772    4796    5059    5083    5345    5369
```

```
           5634    5658    6091    6395    6419    6721    6745    7049    7073    7377    7401    7669    7693    7960      SEQ 1859
           7984    8250    8274    8539    8563    8784    8843    8864
.MSOA       611     615     824     830     959    1071    1112    1118    1120    1332    1334    1336    1358    1375
           1381    1383    1430    1437    1444    1738    1761    1787    1793    1795    1802    1992    2012    2018
           2020    2072    2094    2312    2318    2320    2372    2394    2487    2594    2598    2600    2905    2911
           2913    3215    3221    3223    3514    3516    3540    3546    3548    3837    3839    3863    3869    3871
           4160    4162    4186    4192    4194    4484    4486    4510    4516    4518    4772    4774    4798    4804
           4806    5059    5061    5085    5091    5093    5345    5347    5371    5377    5379    5634    5636    5660
           5666    5668    5774    5778    5780    6093    6099    6101    6395    6397    6421    6427    6429    6721
           6723    6747    6753    6755    7049    7051    7075    7081    7083    7377    7379    7403    7409    7411
           7669    7671    7695    7701    7703    7960    7962    7986    7992    7994    8250    8252    8276    8282
           8284    8539    8541    8565    8571    8573    8805    8819    8823    8845    8851    8853    8866    8872
           8874
.MSOB      1110    1347    1349    1373    1404    1408    1410    1680    1684    1695    1699    1716    1720    1724
           1728    1732    1736    1763    1765    1785    1986    1994    2010    2098    2102    2269    2274    2287
           2292    2310    2398    2402    2899    3209    3534    3857    4180    4504    4792    5079    5365    5654
           6087    6415    6741    7069    7397    7689    7980    8270    8559    8758    8762    8839    8860
.MSAB       275     282     289     298     307     316     325     334     343     352     361     370     379     388
            397     406     415     424     433     442     451     460     469     478     487     496     505     514
            523     532     541     550     559     568     577     586     595     597     599     609     742     749
            756     765     774     783     792     801     810     822    1066    1343    1740    1742    1744    1984
           2272    2903    3213    3538    3861    4184    4508    4796    5083    5369      58    6091    6419    6745
           7073    7401    7693    7984    8274    8563    8780    8843    8864
.MSDO       273     280     287     296     305     314     323     332     341     350     359     368     377     386
            395     404     413     422     431     440     449     458     467     476     485     494     503     512
            521     530     539     548     557     566     575     584     593     604     613     740     747     754
            763     772     781     790     799     808     817     826     828     938     957    1062    1064    1108
           1371    1406    1586    1588    1590    1592    1676    1678    1691    1693    1706    1708    1710    1712
           1783    1981    2008    2092    2096    2280    2308    2392    2396    2485    2596    2859    2874    2897
           3169    3184    3207    3486    3501    3512    3518    3532    3809    3824    3835    3841    3855    4132
           4147    4158    4164    4178    4456    4471    4482    4488    4502    4744    4759    4770    4776    4790
           5031    5046    5057    5063    5077    5317    5332    5343    5349    5363    5606    5621    5632    5638
           5652    5776    6043    6062    6085    6363    6382    6393    6399    6413    6689    6708    6719    6725
           6739    7017    7036    7047    7053    7067    7345    7364    7375    7381    7395    7637    7656    7667
           7673    7687    7928    7947    7958    7964    7978    8218    8237    8248    8254    8268    8507    8526
           8537    8543    8557    8778    8803    8837    8858
.MSELC      943     955     964     971    1076    1088    1100    1127    1356    1365    1392    1404    1421    1428
           1456    1749    1763    1777    1823    1979    2002    2072    2074    2080    2084    2090    2100    2107
           2289    2302    2372    2374    2380    2384    2390    2400    2407    2857    2872    2883    2890    3182
           3193    3200    3499    3510    3525    3822    3833    3848    4145    4156    4171    4469    4480    4495
           4757    4768    4783    5044    5055    5070    5330    5341    5356    5619    5630    5645    6060    6071
           6078    6380    6391    6406    6706    6717    6732    7034    7045    7060    7362    7373    7388    7654
           7665    7680    7945    7956    7971    8235    8246    8261    8524    8535    8550    8801    8812    8830
.MSELE      615     830     959    1112    1114    1118    1120    1358    1375    1377    1381    1383    1430    1437
           1444    1787    1789    1793    1795    1802    1984    1986    1992    1994    2012    2014    2018    2020
           2272    2274    2280    2312    2314    2318    2320    2487    2600    2905    2907    2911    2913    3215
           3217    3221    3223    3540    3542    3546    3548    3863    3865    3869    3871    4186    4188    4192
           4194    4510    4512    4516    4518    4798    4800    4804    4806    5085    5087    5091    5093    5371
           5373    5377    5379    5660    5662    5666    5668    5780    6093    6095    6099    6101    6421    6423
           6427    6429    6747    6749    6753    6755    7075    7077    7081    7083    7403    7405    7409    7411
           7695    7697    7701    7703    7986    7988    7992    7994    8276    8278    8282    8284    8565    8567
           8571    8573    8819    8823    8845    8847    8851    8853    8866    8868    8872    8874
.MSELF      611    1071    1761    2287    2594    3167    3484    3807    4130    4454    4742    5029    5315    5604
```

```
            5774   6041   6056   6058   6361   6376   6378   6687   6702   6704   7015   7030   7032   7343
            7358   7360   7635   7650   7652   7926   7941   7943   8216   8231   8233   8505   8520   8522
            8762   8764   8766   8768   8770   8772   8774   8776   8797   8799
.MSELM       613    824    828    957    973   1129   1406   1451   1458   1818   1825   1981   2070   2082
            2092   2109   2370   2382   2392   2409   2485   2596   2859   2874   2881   2892   3169   3184
            3191   3202   3486   3501   3508   3516   3527   3809   3824   3831   3839   3850   4132   4147
            4154   4162   4173   4456   4471   4478   4486   4497   4744   4759   4766   4774   4785   5031
            5046   5053   5061   5072   5317   5332   5339   5347   5358   5606   5621   5628   5636   5647
            5776   6043   6062   6069   6080   6363   6382   6389   6397   6408   6689   6708   6715   6723
            6734   7017   7036   7043   7051   7062   7345   7364   7371   7379   7390   7637   7656   7663
            7671   7682   7928   7947   7954   7962   7973   8218   8237   8244   8252   8263   8507   8526
            8533   8541   8552   8778   8803   8810   8832
.MSELP       826   3518   3841   4164   4488   4776   5063   5349   5638   6399   6725   7053   7381   7673
            7964   8254   8543
.MSKCN      1108   1371   1586   1588   1590   1592   1678   1693   1706   1708   1710   1712   1783   2008
            2096   2308   2396   3512   3835   4158   4482   4770   5057   5343   5632   6393   6719   7047
            7375   7667   7958   8248   8537
.MSMIN      1765   8780
.MXNOR       275    742   1066   1408   1430
.MXOR       1343
.RA          275    282    289    298    307    316    325    334    343    352    361    370    379    388
             397    406    415    424    433    442    451    460    469    478    487    496    505    514
             523    532    541    550    559    568    577    586    595    597    599    609    611    615
             742    749    756    765    774    783    792    801    810    822    824    830   1112   1118
            1120   1332   1334   1336   1343   1358   1375   1381   1383   1430   1437   1444   1740   1742
            1744   1761   1787   1793   1795   1802   1984   1992   2012   2018   2020   2072   2094   2272
            2312   2318   2320   2372   2394   2903   2905   2911   2913   3213   3215   3221   3223   3538
            3540   3546   3548   3861   3863   3869   3871   4184   4186   4192   4194   4508   4510   4516
            4518   4796   4798   4804   4806   5083   5085   5091   5093   5369   5371   5377   5379   5658
            5660   5666   5668   6091   6093   6099   6101   6419   6421   6427   6429   6745   6747   6753
            6755   7073   7075   7081   7083   7401   7403   7409   7411   7693   7695   7701   7703   7984
            7986   7992   7994   8274   8276   8282   8284   8563   8565   8571   8573   8780   8805   8819
            8823   8843   8845   8851   8853   8864   8866   8872   8874
.RAND        273    280    287    296    305    314    323    332    341    350    359    368    377    386
             395    404    413    422    431    440    449    458    467    476    485    494    503    512
             521    530    539    548    557    566    575    584    593    604    740    747    754    763
             772    781    790    799    808    817    938   1062   1064   1334   1404   1410   1676   1691
            2102   2269   2402   2899   3207   3532   3855   4178   4502   4790   5077   5363   5652   6085
            6413   6739   7067   7395   7687   7978   8268   8557   8758   8837   8858
.RB          273    275    280    282    287    289    296    298    305    307    314    316    323    325
             332    334    341    343    350    352    359    361    368    370    377    379    386    388
             395    397    404    406    413    415    422    424    431    433    440    442    449    451
             458    460    467    469    476    478    485    487    494    496    503    505    512    514
             521    523    530    532    539    541    548    550    557    559    566    568    575    577
             584    586    593    595    597    599    604    609    613    740    742    747    749    754
             756    763    765    772    774    781    783    790    792    799    801    808    810    817
             822    828   1108   1110   1332   1334   1336   1343   1347   1349   1371   1373   1404   1406
            1408   1410   1588   1590   1592   1676   1678   1680   1684   1691   1693   1695   1699   1708
            1710   1712   1716   1720   1724   1728   1732   1736   1738   1740   1742   1744   1763   1765
            1783   1785   1981   1984   1986   1994   2008   2010   2072   2092   2094   2096   2098   2102
            2269   2272   2274   2280   2287   2292   2308   2310   2372   2392   2394   2396   2398   2402
            2859   2874   2897   2899   2903   3169   3184   3207   3209   3213   3486   3501   3532   3534
            3538   3809   3824   3855   3857   3861   4132   4147   4178   4180   4184   4456   4471   4502
```

```
          4504  4508  4744  4759  4790  4792  4796  5031  5046  5077  5079  5083  5317  5332
          5363  5365  5369  5606  5621  5652  5654  5658  6043  6062  6085  6087  6091  6363
          6382  6413  6415  6419  6689  6708  6739  6741  6745  7017  7036  7067  7069  7073
          7345  7364  7395  7397  7401  7637  7656  7687  7689  7693  7928  7947  7978  7980
          7984  8218  8237  8268  8270  8274  8507  8526  8557  8559  8563  8758  8762  8778
          8780  8803  8805  8837  8839  8843  8858  8860  8864
.RBAD     2602  5782
.RCCCA     948  1081  1093  1398  1755  1975  2086  2283  2386  2868  3178  3495  3818  4141
          4465  4753  5040  5326  5615  6052  6372  6698  7026  7354  7646  7937  8227  8516
          8793
.RCCCE    1412  1751
.RCCCP    1414  1753
.RCCER    1988  2276
.RCCFZ    1343  1765  2598  5778  8780
.RCCGC    1114  1377  1789  2014  2314  2907  3217  3542  3865  4188  4512  4800  5087  5373
          5662  6095  6423  6749  7077  7405  7697  7988  8278  8567  8847  8868
.RCCLW    1408
.RCCMB    1349
.RCENA     948  1081  1093  1114  1343  1349  1377  1398  1408  1412  1414  1751  1753  1755
          1765  1789  1975  1988  2014  2086  2276  2283  2314  2386  2598  2868  2907  3178
          3217  3495  3542  3818  3865  4141  4188  4465  4512  4753  4800  5040  5087  5326
          5373  5615  5662  5778  6052  6095  6372  6423  6698  6749  7026  7077  7354  7405
          7646  7697  7937  7988  8227  8278  8516  8567  8780  8793  8847  8868
.RCJP      948  1081  1093  1114  1343  1349  1377  1398  1408  1412  1414  1751  1753  1755
          1765  1789  1975  1988  2014  2086  2276  2283  2314  2386  2598  2868  2907  3178
          3217  3495  3542  3818  3865  4141  4188  4465  4512  4753  4800  5040  5087  5326
          5373  5615  5662  5778  6052  6095  6372  6423  6698  6749  7026  7077  7354  7405
          7646  7697  7937  7988  8227  8278  8516  8567  8780  8793  8847  8868
.RCJS     1330  1341  1749  1994  2289
.RCONT     273   280   287   296   305   314   323   332   341   350   359   368   377   386
           395   404   413   422   431   440   449   458   467   476   485   494   503   512
           521   530   539   548   557   566   575   584   593   611   613   615   740   747
           754   763   772   781   790   799   808   824   826   828   830   943   955   957
           959   964  1071  1076  1088  1100  1112  1118  1120  1332  1334  1336  1347  1356
          1358  1365  1375  1381  1383  1392  1402  1404  1406  1410  1421  1428  1430  1588
          1590  1592  1708  1710  1712  1738  1740  1742  1744  1759  1761  1763  1777  1787
          1793  1795  1979  1981  1984  1986  1992  2002  2012  2018  2020  2027  2029  2031
          2033  2035  2037  2039  2041  2043  2045  2047  2049  2051  2053  2055  2057  2059
          2061  2063  2090  2092  2094  2100  2272  2274  2280  2287  2302  2312  2318  2320
          2327  2329  2331  2333  2335  2337  2339  2341  2343  2345  2347  2349  2351  2353
          2355  2357  2359  2361  2363  2372  2390  2392  2394  2400  2596  2600  2857  2859
          2872  2874  2897  2899  2905  2911  3167  3169  3182  3184  3207  3209  3215  3221
          3484  3486  3499  3501  3510  3516  3532  3534  3540  3546  3807  3809  3822  3824
          3833  3839  3855  3857  3863  3869  4130  4132  4145  4147  4156  4162  4178  4180
          4186  4192  4454  4456  4469  4471  4480  4486  4502  4504  4510  4516  4742  4744
          4757  4759  4768  4774  4790  4792  4798  4804  5029  5031  5044  5046  5055  5061
          5077  5079  5085  5091  5315  5317  5330  5332  5341  5347  5363  5365  5371  5377
          5604  5606  5619  5621  5630  5636  5652  5654  5660  5666  5776  5780  6041  6043
          6056  6058  6060  6062  6085  6087  6093  6099  6361  6363  6376  6378  6380  6382
          6391  6397  6413  6415  6421  6427  6687  6689  6702  6704  6706  6708  6717  6723
          6739  6741  6747  6753  7015  7017  7030  7032  7034  7036  7045  7051  7067  7069
          7075  7081  7343  7345  7358  7360  7362  7364  7373  7379  7395  7397  7403  7409
          7635  7637  7650  7652  7654  7656  7665  7671  7687  7689  7695  7701  7926  7928
```

```
            7941    7943    7945    7947    7956    7962    7978    7980    7986    7992    8216    8218    8231    8233
            8235    8237    8246    8252    8268    8270    8276    8282    8505    8507    8520    8522    8524    8526
            8535    8541    8557    8559    8565    8571    8762    8764    8766    8768    8770    8772    8774    8776
            8778    8797    8799    8801    8803    8837    8839    8845    8851    8858    8860    8866    8872
.RCRTN      1416    1813    2065    2365
.RCRY        282     289     298     307     316     325     334     343     352     361     370     379     388     397
             406     415     424     433     442     451     460     469     478     487     496     505     514     523
             532     541     550     559     568     577     586     595     749     756     765     774     783     792
             801     810    1336    1347    1763    1986    1994    2274    2292    2899    3209    3534    3857    4180
            4504    4792    5079    5365    5654    6087    6415    6741    7069    7397    7689    7980    8270    8559
            8762    8780    8839    8860
.RD          273     275     280     282     287     289     296     298     305     307     314     316     323     325
             332     334     341     343     350     352     359     361     368     370     377     379     386     388
             395     397     404     406     413     415     422     424     431     433     440     442     449     451
             458     460     467     469     476     478     485     487     494     496     503     505     512     514
             521     523     530     532     539     541     548     550     557     559     566     568     575     577
             584     586     593     595     597     599     604     609     611     613     615     740     742     747
             749     754     756     763     765     772     774     781     783     790     792     799     801     808
             810     817     822     824     826     828     830     938     943     948     950     955     957     959
            1062    1064    1066    1071    1075    1081    1083    1088    1093    1095    1108    1110    1112    1118
            1120    1330    1332    1334    1336    1341    1343    1347    1349    1358    1371    1373    1375    1381
            1383    1404    1406    1408    1410    1430    1437    1444    1586    1588    1590    1592    1676    1678
            1680    1682    1684    1691    1693    1695    1697    1699    1706    1708    1710    1712    1714    1716
            1718    1720    1722    1724    1726    1728    1730    1732    1734    1736    1738    1740    1742    1744
            1761    1763    1765    1783    1785    1787    1793    1795    1802    1809    1811    1981    1984    1986
            1992    1994    2008    2010    2012    2018    2020    2072    2092    2094    2096    2098    2102    2269
            2272    2274    2280    2287    2289    2292    2308    2310    2312    2318    2320    2372    2392    2394
            2396    2398    2402    2485    2487    2594    2596    2598    2600    2855    2857    2859    2861    2868
            2870    2872    2874    2883    2897    2899    2903    2905    2907    2909    2911    2913    3165    3167
            3169    3171    3178    3180    3182    3184    3193    3207    3209    3213    3215    3217    3219    3221
            3223    3482    3484    3486    3488    3495    3497    3499    3501    3510    3512    3514    3516    3518
            3532    3534    3538    3540    3542    3544    3546    3548    3805    3807    3809    3811    3818    3820
            3822    3824    3833    3835    3837    3839    3841    3855    3857    3861    3863    3865    3867    3869
            3871    4128    4130    4132    4134    4141    4143    4145    4147    4156    4158    4160    4162    4164
            4178    4180    4184    4186    4188    4190    4192    4194    4452    4454    4456    4458    4465    4467
            4469    4471    4480    4482    4484    4486    4488    4502    4504    4508    4510    4512    4514    4516
            4518    4740    4742    4744    4746    4753    4755    4757    4759    4768    4770    4772    4774    4776
            4790    4792    4796    4798    4800    4802    4804    4806    5027    5029    5031    5033    5040    5042
            5044    5046    5055    5057    5059    5061    5063    5077    5079    5083    5085    5087    5089    5091
            5093    5313    5315    5317    5319    5326    5328    5330    5332    5341    5343    5345    5347    5349
            5363    5365    5369    5371    5373    5375    5377    5379    5602    5604    5606    5608    5615    5617
            5619    5621    5630    5632    5634    5636    5638    5652    5654    5658    5660    5662    5664    5666
            5668    5774    5776    5778    5780    6039    6041    6043    6045    6052    6054    6056    6058    6060
            6062    6071    6085    6087    6091    6093    6095    6097    6099    6101    6359    6361    6363    6365
            6372    6374    6376    6378    6380    6382    6391    6393    6395    6397    6399    6413    6415    6419
            6421    6423    6425    6427    6429    6685    6687    6689    6691    6698    6700    6702    6704    6706
            6708    6717    6719    6721    6723    6725    6739    6741    6745    6747    6749    6751    6753    6755
            7013    7015    7017    7019    7026    7028    7030    7032    7034    7036    7045    7047    7049    7051
            7053    7067    7069    7073    7075    7077    7079    7081    7083    7341    7343    7345    7347    7354
            7356    7358    7360    7362    7364    7373    7375    7377    7379    7381    7395    7397    7401    7403
            7405    7407    7409    7411    7633    7635    7637    7639    7646    7648    7650    7652    7654    7656
            7665    7667    7669    7671    7673    7687    7689    7693    7695    7697    7699    7701    7703    7924
            7926    7928    7930    7937    7939    7941    7943    7945    7947    7956    7958    7960    7962    7964
```

```
Q 1850                                                                                                                                  SEQ 1863
            7978     7980     7984     7986     7988     7990     7992     7994     8214     8216     8218     8220     8227     8229
            8231     8233     8235     8237     8246     8248     8250     8252     8254     8268     8270     8274     8276     8278
            8280     8282     8284     8503     8505     8507     8509     8516     8518     8520     8522     8524     8526     8535
            8537     8539     8541     8543     8557     8559     8563     8565     8567     8569     8571     8573     8758     8762
            8778     8780     8782     8786     8793     8795     8797     8799     8801     8803     8805     8812     8819     8823
            8837     8839     8843     8845     8847     8849     8851     8853     8858     8860     8864     8866     8868     8870
            8872     8874
.RJ          194      196      198      200      202      204      206      208      210      212      214      216      218      220
             222      224      226      228      230      232      234      236      238      240      242      244      246      248
             250      252      254      256      258      260      262      264      266      268      275      282      289      291
             298      300      307      309      316      318      325      327      334      336      343      345      352      354
             361      363      370      372      379      381      388      390      397      399      406      408      415      417
             424      426      433      435      442      444      451      453      460      462      469      471      478      480
             487      489      496      498      505      507      514      516      523      525      532      534      541      543
             550      552      559      561      568      570      577      579      586      588      595      597      599      604
             609      617      717      719      721      723      725      727      729      731      733      735      742      749
             756      758      765      767      774      776      783      785      792      794      801      803      810      812
             817      822      832      938      948      950      966      971      973     1062     1064     1066     1081     1083
            1093     1095     1102     1108     1110     1114     1116     1122     1127     1129     1330     1341     1343     1345
            1349     1351     1360     1371     1373     1377     1379     1385     1394     1396     1398     1400     1408     1412
            1414     1423     1432     1437     1439     1444     1446     1456     1458     1586     1676     1678     1680     1682
            1684     1691     1693     1695     1697     1699     1706     1714     1716     1718     1720     1722     1724     1726
            1728     1730     1732     1734     1736     1749     1751     1753     1755     1757     1765     1767     1773     1775
            1783     1785     1789     1791     1797     1802     1804     1809     1811     1823     1825     1975     1988     1990
            1994     1997     2008     2010     2014     2016     2022     2070     2072     2080     2082     2084     2086     2088
            2096     2098     2102     2107     2109     2269     2276     2278     2283     2285     2289     2292     2298     2300
            2308     2310     2314     2316     2322     2370     2380     2382     2384     2386     2388     2396     2398     2402
            2407     2409     2485     2487     2594     2598     2602     2855     2861     2866     2868     2870     2881     2890
            2901     2903     2907     2909     2913     3165     3171     3176     3178     3180     3191     3200     3211     3213
            3217     3219     3223     3482     3488     3493     3495     3497     3508     3512     3514     3525     3536     3538
            3542     3544     3548     3805     3811     3816     3818     3820     3831     3835     3837     3848     3859     3861
            3865     3867     3871     4128     4134     4139     4141     4143     4154     4158     4160     4171     4182     4184
            4188     4190     4194     4452     4458     4463     4465     4467     4478     4482     4484     4495     4506     4508
            4512     4514     4518     4740     4746     4751     4753     4755     4766     4770     4772     4783     4794     4796
            4800     4802     4806     5027     5033     5038     5040     5042     5053     5057     5059     5070     5081     5083
            5087     5089     5093     5313     5319     5324     5326     5328     5339     5343     5345     5356     5367     5369
            5373     5375     5379     5602     5608     5613     5615     5617     5628     5632     5634     5645     5656     5658
            5662     5664     5668     5774     5778     5782     6039     6045     6050     6052     6054     6069     6078     6089
            6091     6095     6097     6101     6359     6365     6370     6372     6374     6389     6393     6395     6406     6417
            6419     6423     6425     6429     6685     6691     6696     6698     6700     6715     6719     6721     6732     6743
            6745     6749     6751     6755     7013     7019     7024     7026     7028     7043     7047     7049     7060     7071
            7073     7077     7079     7083     7341     7347     7352     7354     7356     7371     7375     7377     7388     7399
            7401     7405     7407     7411     7633     7639     7644     7646     7648     7663     7667     7669     7680     7691
            7693     7697     7699     7703     7924     7930     7935     7937     7939     7954     7958     7960     7971     7982
            7984     7988     7990     7994     8214     8220     8225     8227     8229     8244     8248     8250     8261     8272
            8274     8278     8280     8284     8503     8509     8514     8516     8518     8533     8537     8539     8550     8561
            8563     8567     8569     8573     8758     8760     8780     8782     8784     8786     8791     8793     8795     8610
            8819     8821     8823     8825     8830     8841     8843     8847     8849     8853     8862     8864     8868     8870
            8874
.RJMAP       194      196      198      200      202      204      206      208      210      212      214      216      218      220
             222      224      226      228      230      232      234      236      238      240      242      244      246      248
             250      252      254      256      258      260      262      264      266      268      275      282      291      300
             309      318      327      336      345      354      363      372      381      390      399      408      417      426
```

```
          435   444   453   462   471   480   489   498   507   516   525   534   543   552
          561   570   579   588   599   604   617   717   719   721   723   725   727   729
          731   733   735   742   749   758   767   776   785   794   803   812   817   832
          938   950   966   971   973  1062  1064  1066  1083  1095  1102  1116  1122  1127
         1129  1345  1351  1360  1379  1385  1400  1423  1432  1437  1439  1444  1446  1456
         1458  1586  1676  1691  1706  1757  1767  1791  1797  1802  1804  1823  1825  1990
         2016  2022  2070  2080  2082  2084  2088  2102  2107  2109  2278  2285  2316  2322
         2370  2380  2382  2384  2388  2402  2407  2409  2485  2487  2594  2602  2855  2861
         2866  2870  2881  2890  2909  2913  3165  3171  3176  3180  3191  3200  3219  3223
         3482  3488  3493  3497  3508  3525  3544  3548  3805  3811  3816  3820  3831  3848
         3867  3871  4128  4134  4139  4143  4154  4171  4190  4194  4452  4458  4463  4467
         4478  4495  4514  4518  4740  4746  4751  4755  4766  4783  4802  4806  5027  5033
         5038  5042  5053  5070  5089  5093  5313  5319  5324  5328  5339  5356  5375  5379
         5602  5608  5613  5617  5628  5645  5664  5668  5774  5782  6039  6045  6050  6054
         6069  6078  6097  6101  6359  6365  6370  6374  6389  6406  6425  6429  6685  6691
         6696  6700  6715  6732  6751  6755  7013  7019  7024  7028  7043  7060  7079  7083
         7341  7347  7352  7356  7371  7388  7407  7411  7633  7639  7644  7648  7663  7680
         7699  7703  7924  7930  7935  7939  7954  7971  7990  7994  8214  8220  8225  8229
         8244  8261  8280  8284  8503  8509  8514  8518  8533  8550  8569  8573  8758  8782
         8786  8791  8795  8810  8819  8821  8823  8825  8830  8849  8853  8870  8874
.RJZ     1451  1686  1701  1818  1977  2074  2374  2876  2883  2885  2892  3186  3193  3195
         3202  3503  3518  3520  3527  3826  3841  3843  3850  4149  4164  4166  4173  4473
         4488  4490  4497  4761  4776  4778  4785  5048  5063  5065  5072  5334  5349  5351
         5358  5623  5638  5640  5647  6064  6071  6073  6080  6384  6399  6401  6408  6710
         6725  6727  6734  7038  7053  7055  7062  7366  7381  7383  7390  7658  7673  7675
         7682  7949  7964  7966  7973  8239  8254  8256  8263  8528  8543  8545  8552  8805
         8812  8814  8832
.RLDCT    289   298   307   316   325   334   343   352   361   370   379   388   397   406
          415   424   433   442   451   460   469   478   487   496   505   514   523   532
          541   550   559   568   577   586   595   756   765   774   783   792   801   810
         1108  1371  1394  1678  1682  1693  1697  1714  1718  1722  1726  1730  1734  1773
         1783  1809  2008  2072  2096  2269  2298  2308  2396  2901  3211  3512  3536  3835
         3859  4158  4182  4482  4506  4770  4794  5057  5081  5343  5367  5632  5656  6089
         6393  6417  6719  6743  7047  7071  7375  7399  7667  7691  7958  7982  8248  8272
         8537  8561  8760  8841  8862
.RMGC     611   613   615   824   826   828   830   943   955   957   959   964   971   973
         1071  1076  1088  1100  1108  1112  1114  1118  1120  1127  1129  1356  1358  1365
         1371  1375  1377  1381  1383  1392  1404  1406  1421  1428  1430  1437  1444  1451
         1456  1458  1586  1588  1590  1592  1678  1693  1706  1708  1710  1712  1749  1761
         1763  1777  1783  1787  1789  1793  1795  1802  1818  1823  1825  1979  1981  1984
         1986  1992  1994  2002  2008  2012  2014  2018  2020  2070  2072  2074  2080  2082
         2084  2090  2092  2096  2100  2107  2109  2272  2274  2280  2287  2289  2302  2308
         2312  2314  2318  2320  2370  2372  2374  2380  2382  2384  2390  2392  2396  2400
         2407  2409  2485  2487  2594  2596  2600  2857  2859  2872  2874  2881  2883  2890
         2892  2905  2907  2911  2913  3167  3169  3182  3184  3191  3193  3200  3202  3215
         3217  3221  3223  3484  3486  3499  3501  3508  3510  3512  3516  3518  3525  3527
         3540  3542  3546  3548  3807  3809  3822  3824  3831  3833  3835  3839  3841  3848
         3850  3863  3865  3869  3871  4130  4132  4145  4147  4154  4156  4158  4162  4164
         4171  4173  4186  4188  4192  4194  4454  4456  4469  4471  4478  4480  4482  4486
         4488  4495  4497  4510  4512  4516  4518  4742  4744  4757  4759  4766  4768  4770
         4774  4776  4783  4785  4798  4800  4804  4806  5029  5031  5044  5046  5053  5055
         5057  5061  5063  5070  5072  5085  5087  5091  5093  5315  5317  5330  5332  5339
         5341  5343  5347  5349  5356  5358  5371  5373  5377  5379  5604  5606  5619  5621
```

```
          5628  5630  5632  5636  5638  5645  5647  5660  5662  5666  5668  5774  5776  5780
          6041  6043  6056  6058  6060  6062  6069  6071  6078  6080  6093  6095  6099  6101
          6361  6363  6376  6378  6380  6382  6389  6391  6393  6397  6399  6406  6408  6421
          6423  6427  6429  6687  6689  6702  6704  6706  6708  6715  6717  6719  6723  6725
          6732  6734  6747  6749  6753  6755  7015  7017  7030  7032  7034  7036  7043  7045
          7047  7051  7053  7060  7062  7075  7077  7081  7083  7343  7345  7358  7360  7362
          7364  7371  7373  7375  7379  7381  7388  7390  7403  7405  7409  7411  7635  7637
          7650  7652  7654  7656  7663  7665  7667  7671  7673  7680  7682  7695  7697  7701
          7703  7926  7928  7941  7943  7945  7947  7954  7956  7958  7962  7964  7971  7973
          7986  7988  7992  7994  8216  8218  8231  8233  8235  8237  8244  8246  8248  8252
          8254  8261  8263  8276  8278  8282  8284  8505  8507  8520  8522  8524  8526  8533
          8535  8537  8541  8543  8550  8552  8565  8567  8571  8573  8762  8764  8766  8768
          8770  8772  8774  8776  8778  8797  8799  8801  8803  8810  8812  8819  8823  8830
          8832  8845  8847  8851  8853  8866  8868  8872  8874
.ROENA     611   615   824   830   959  1071  1112  1120  1358  1375  1383  1430  1437  1444
          1761  1787  1795  1802  1984  1992  2012  2020  2272  2287  2312  2320  2487  2600
          2905  2913  3215  3223  3516  3540  3548  3839  3863  3871  4162  4186  4194  4486
          4510  4518  4774  4798  4806  5061  5085  5093  5347  5371  5379  5636  5660  5668
          5780  6093  6101  6397  6421  6429  6723  6747  6755  7051  7075  7083  7379  7403
          7411  7671  7695  7703  7962  7986  7994  8252  8276  8284  8541  8565  8573  8819
          8823  8845  8853  8866  8874
.ROR       824   828   957   959  1108  1110  1112  1118  1120  1332  1349  1358  1371  1373
          1375  1381  1383  1406  1437  1444  1586  1588  1590  1592  1678  1680  1684  1693
          1695  1699  1706  1708  1710  1712  1716  1720  1724  1728  1732  1736  1738  1740
          1742  1744  1783  1785  1787  1793  1795  1802  1981  1984  1992  2008  2010  2012
          2018  2020  2072  2092  2094  2096  2098  2272  2280  2287  2308  2310  2312  2318
          2320  2372  2392  2394  2396  2398  2485  2487  2596  2598  2600  2859  2874  2905
          2911  2913  3169  3184  3215  3221  3223  3486  3501  3512  3514  3516  3540  3546
          3548  3809  3824  3835  3837  3839  3863  3869  3871  4132  4147  4158  4160  4162
          4186  4192  4194  4456  4471  4482  4484  4486  4510  4516  4518  4744  4759  4770
          4772  4774  4798  4804  4806  5031  5046  5057  5059  5061  5085  5091  5093  5317
          5332  5343  5345  5347  5371  5377  5379  5606  5621  5632  5634  5636  5660  5666
          5668  5776  5778  5780  6043  6062  6093  6099  6101  6363  6382  6393  6395  6397
          6421  6427  6429  6689  6708  6719  6721  6723  6747  6753  6755  7017  7036  7047
          7049  7051  7075  7081  7083  7345  7364  7375  7377  7379  7403  7409  7411  7637
          7656  7667  7669  7671  7695  7701  7703  7928  7947  7958  7960  7962  7986  7992
          7994  8218  8237  8248  8250  8252  8276  8282  8284  8507  8526  8537  8539  8541
          8565  8571  8573  8778  8803  8805  8819  8823  8845  8851  8853  8866  8872  8874
.RPLUS     282   289   298   307   316   325   334   343   352   361   370   379   388   397
           406   415   424   433   442   451   460   469   478   487   496   505   514   523
           532   541   550   559   568   577   586   595   597   609   749   756   765   774
           783   792   801   810   822  1336  1347  1763  1986  1994  2274  2292  2899  2903
          3209  3213  3534  3538  3857  3861  4180  4184  4504  4508  4792  4796  5079  5083
          5365  5369  5654  5658  6087  6091  6415  6419  6741  6745  7069  7073  7397  7401
          7689  7693  7980  7984  8270  8274  8559  8563  8762  8839  8843  8860  8864
.RRPCT     597   609   822  1110  1373  1396  1680  1684  1695  1699  1716  1720  1724  1728
          1732  1736  1775  1785  1811  1997  2010  2098  2292  2300  2310  2398  2903  3213
          3514  3538  3837  3861  4160  4184  4484  4508  4772  4796  5059  5083  5345  5369
          5634  5658  6091  6395  6419  6721  6745  7049  7073  7377  7401  7669  7693  7960
          7984  8250  8274  8539  8563  8784  8843  8864
.RSOA      611   615   824   830   959  1071  1112  1118  1120  1332  1334  1336  1358  1375
          1381  1383  1430  1437  1444  1738  1761  1787  1793  1795  1802  1992  2012  2018
          2020  2072  2094  2312  2318  2320  2372  2394  2487  2594  2598  2600  2905  2911
          2913  3215  3221  3223  3514  3516  3540  3546  3548  3837  3839  3863  3869  3871
          4160  4162  4186  4192  4194  4484  4486  4510  4516  4518  4772  4774  4798  4804
          4806  5059  5061  5085  5091  5093  5345  5347  5371  5377  5379  5634  5636  5660
          5666  5668  5774  5778  5780  6093  6099  6101  6395  6397  6421  6427  6429  6721
          6723  6747  6753  6755  7049  7051  7075  7081  7083  7377  7379  7403  7409  7411
          7669  7671  7695  7701  7703  7960  7962  7986  7992  7994  8250  8252  8276  8282
          8284  8539  8541  8565  8571  8573  8805  8819  8823  8845  8851  8853  8866  8872
          8874
.RSOB     1110  1347  1349  1373  1404  1408  1410  1680  1684  1695  1699  1716  1720  1724
          1728  1732  1736  1763  1765  1785  1986  1994  2010  2098  2102  2269  2274  2287
          2292  2310  2398  2402  2899  3209  3534  3857  4180  4504  4792  5079  5365  5654
          6087  6415  6741  7069  7397  7689  7980  8270  8559  8758  8762  8839  8860
.RSAB      275   282   289   298   307   316   325   334   343   352   361   370   379   388
           397   406   415   424   433   442   451   460   469   478   487   496   505   514
           523   532   541   550   559   568   577   586   595   597   599   609   742   749
           756   765   774   783   792   801   810   822  1066  1343  1740  1742  1744  1984
          2272  2903  3213  3538  3861  4184  4508  4796  5083  5369  5658  6091  6419  6745
          7073  7401  7693  7984  8274  8563  8780  8843  8864
.RSDO      273   280   287   296   305   314   323   332   341   350   359   368   377   386
           395   404   413   422   431   440   449   458   467   476   485   494   503   512
           521   530   539   548   557   566   575   584   593   604   613   740   747   754
           763   772   781   790   799   808   817   826   828   938   957  1062  1064  1108
          1371  1406  1586  1588  1590  1592  1676  1678  1691  1693  1706  1708  1710  1712
          1783  1981  2008  2092  2096  2280  2308  2392  2396  2485  2596  2859  2874  2897
          3169  3184  3207  3486  3501  3512  3518  3532  3809  3824  3835  3841  3855  4132
          4147  4158  4164  4178  4456  4471  4482  4488  4502  4744  4759  4770  4776  4790
          5031  5046  5057  5063  5077  5317  5332  5343  5349  5363  5606  5621  5632  5638
          5652  5776  6043  6062  6085  6363  6382  6393  6399  6413  6689  6708  6719  6725
          6739  7017  7036  7047  7053  7067  7345  7364  7375  7381  7395  7637  7656  7667
          7673  7687  7928  7947  7958  7964  7978  8218  8237  8248  8254  8268  8507  8526
          8537  8543  8557  8778  8803  8837  8858
.RSELC     943   955   964   971  1076  1088  1100  1127  1356  1365  1392  1404  1421  1428
          1456  1749  1763  1777  1823  1979  2002  2072  2074  2080  2084  2090  2100  2107
          2289  2302  2372  2374  2380  2384  2390  2400  2407  2857  2872  2883  2890  3182
          3193  3200  3499  3510  3525  3822  3833  3848  4145  4156  4171  4469  4480  4495
          4757  4768  4783  5044  5055  5070  5330  5341  5356  5619  5630  5645  6060  6071
          6078  6380  6391  6406  6706  6717  6732  7034  7045  7060  7362  7373  7388  7654
          7665  7680  7945  7956  7971  8235  8246  8261  8524  8535  8550  8801  8812  8830
.RSELE     615   830   959  1112  1114  1118  1120  1358  1375  1377  1381  1383  1430  1437
          1444  1787  1789  1793  1795  1802  1984  1986  1992  1994  2012  2014  2018  2020
          2272  2274  2280  2312  2314  2318  2320  2487  2600  2905  2907  2911  2913  3215
          3217  3221  3223  3540  3542  3546  3548  3863  3865  3869  3871  4186  4188  4192
          4194  4510  4512  4516  4518  4798  4800  4804  4806  5085  5087  5091  5093  5371
          5373  5377  5379  5660  5662  5666  5668  5780  6093  6095  6099  6101  6421  6423
          6427  6429  6747  6749  6753  6755  7075  7077  7081  7083  7403  7405  7409  7411
          7695  7697  7701  7703  7986  7988  7992  7994  8276  8278  8282  8284  8565  8567
          8571  8573  8819  8823  8845  8847  8851  8853  8866  8868  8872  8874
.RSELF     611  1071  1761  2287  2594  3167  3484  3807  4130  4454  4742  5029  5315  5604
          5774  6041  6056  6058  6361  6376  6378  6687  6702  6704  7015  7030  7032  7343
          7358  7360  7635  7650  7652  7926  7941  7943  8216  8231  8233  8505  8520  8522
          8762  8764  8766  8768  8770  8772  8774  8776  8797  8799
.RSELM     613   824   828   957   973  1129  1406  1451  1458  1818  1825  1981  2070  2082
          2092  2109  2370  2382  2392  2409  2485  2596  2859  2874  2881  2892  3169  3184
          3191   3202   3486   3501   3508   3516   3527   3809   3824   3831   3839   3850   4132   4147
          4154   4162   4173   4456   4471   4478   4486   4497   4744   4759   4766   4774   4785   5031
          5046   5053   5061   5072   5317   5332   5339   5347   5358   5606   5621   5628   5636   5647
          5776   6043   6062   6069   6080   6363   6382   6389   6397   6408   6689   6708   6715   6723
          6734   7017   7036   7043   7051   7062   7345   7364   7371   7379   7390   7637   7656   7663
          7671   7682   7928   7947   7954   7962   7973   8218   8237   8244   8252   8263   8507   8526
          8533   8541   8552   8778   8803   8810   8832
.RSELP     826   3518   3841   4164   4488   4776   5063   5349   5638   6399   6725   7053   7381   7673
          7964   8254   8543
.RSKCN    1108   1371   1586   1588   1590   1592   1678   1693   1706   1708   1710   1712   1783   2008
          2096   2308   2396   3512   3835   4158   4482   4770   5057   5343   5632   6393   6719   7047
          7375   7667   7958   8248   8537
.RSMIN    1765   8780
.RXNOR     275    742   1066   1408   1430
.RXOR     1343
```

Q 1855

```
ATABLE      109    111    113    115    117    119    121    123    125    127    129    131    133    135     SEQ 1868
            137    139    141    143    145    147    149    151    153    155    157    159    161    163
            165    167    169    171    173    175    177    179    181    183    185    688    690    692
            694    696    698    700    702    704    706    708   2474   2476
CALC        194    196    198    200    202    204    206    208    210    212    214    216    218    220
            222    224    226    228    230    232    234    236    238    240    242    244    246    248
            250    252    254    256    258    260    262    264    266    268    273    275    280    282
            287    289    291    296    298    300    305    307    309    314    316    318    323    325
            327    332    334    336    341    343    345    350    352    354    359    361    363    368
            370    372    377    379    381    386    388    390    395    397    399    404    406    408
            413    415    417    422    424    426    431    433    435    440    442    444    449    451
            453    458    460    462    467    469    471    476    478    480    485    487    489    494
            496    498    503    505    507    512    514    516    521    523    525    530    532    534
            539    541    543    548    550    552    557    559    561    566    568    570    575    577
            579    584    586    588    593    595    597    599    604    609    611    613    615    617
            717    719    721    723    725    727    729    731    733    735    740    742    747    749
            754    756    758    763    765    767    772    774    776    781    783    785    790    792
            794    799    801    803    808    810    812    817    822    824    826    828    830    832
            938    943    948    950    955    957    959    964    966    971    973   1062   1064   1066
           1071   1076   1081   1083   1088   1093   1095   1100   1102   1108   1110   1112   1114   1116
           1118   1120   1122   1127   1129   1330   1332   1334   1336   1341   1343   1345   1347   1349
           1351   1356   1358   1360   1365   1371   1373   1375   1377   1379   1381   1383   1385   1392
           1394   1396   1398   1400   1402   1404   1406   1408   1410   1412   1414   1416   1421   1423
           1428   1430   1432   1437   1439   1444   1446   1451   1456   1458   1586   1588   1590   1592
           1676   1678   1680   1682   1684   1686   1691   1693   1695   1697   1699   1701   1706   1708
           1710   1712   1714   1716   1718   1720   1722   1724   1726   1728   1730   1732   1734   1736
           1738   1740   1742   1744   1749   1751   1753   1755   1757   1759   1761   1763   1765   1767
           1773   1775   1777   1783   1785   1787   1789   1791   1793   1795   1797   1802   1804   1809
           1811   1813   1818   1823   1825   1975   1977   1979   1981   1984   1986   1988   1990   1992
           1994   1997   2002   2008   2010   2012   2014   2016   2018   2020   2022   2027   2029   2031
           2033   2035   2037   2039   2041   2043   2045   2047   2049   2051   2053   2055   2057   2059
           2061   2063   2065   2070   2072   2074   2080   2082   2084   2086   2088   2090   2092   2094
           2096   2098   2100   2102   2107   2109   2269   2272   2274   2276   2278   2280   2283   2285
           2287   2289   2292   2298   2300   2302   2308   2310   2312   2314   2316   2318   2320   2322
           2327   2329   2331   2333   2335   2337   2339   2341   2343   2345   2347   2349   2351   2353
           2355   2357   2359   2361   2363   2365   2370   2372   2374   2380   2382   2384   2386   2388
           2390   2392   2394   2396   2398   2400   2402   2407   2409   2485   2487   2594   2596   2598
           2600   2602   2855   2857   2859   2861   2866   2868   2870   2872   2874   2876   2881   2883
           2885   2890   2892   2897   2899   2901   2903   2905   2907   2909   2911   2913   3165   3167
           3169   3171   3176   3178   3180   3182   3184   3186   3191   3193   3195   3200   3202   3207
           3209   3211   3213   3215   3217   3219   3221   3223   3482   3484   3486   3488   3493   3495
           3497   3499   3501   3503   3508   3510   3512   3514   3516   3518   3520   3525   3527   3532
           3534   3536   3538   3540   3542   3544   3546   3548   3805   3807   3809   3811   3816   3818
           3820   3822   3824   3826   3831   3833   3835   3837   3839   3841   3843   3848   3850   3855
           3857   3859   3861   3863   3865   3867   3869   3871   4128   4130   4132   4134   4139   4141
           4143   4145   4147   4149   4154   4156   4158   4160   4162   4164   4166   4171   4173   4178
           4180   4182   4184   4186   4188   4190   4192   4194   4452   4454   4456   4458   4463   4465
           4467   4469   4471   4473   4478   4480   4482   4484   4486   4488   4490   4495   4497   4502
           4504   4506   4508   4510   4512   4514   4516   4518   4740   4742   4744   4746   4751   4753
           4755   4757   4759   4761   4766   4768   4770   4772   4774   4776   4778   4783   4785   4790
           4792   4794   4796   4798   4800   4802   4804   4806   5027   5029   5031   5033   5038   5040
           5042   5044   5046   5048   5053   5055   5057   5059   5061   5063   5065   5070   5072   5077
           5079   5081   5083   5085   5087   5089   5091   5093   5313   5315   5317   5319   5324   5326
```

```
Q 1856    5328  5330  5332  5334  5339  5341  5343  5345  5347  5349  5351  5356  5358  5363   SEQ 1869
          5365  5367  5369  5371  5373  5375  5377  5379  5602  5604  5606  5608  5613  5615
          5617  5619  5621  5623  5628  5630  5632  5634  5636  5638  5640  5645  5647  5652
          5654  5656  5658  5660  5662  5664  5666  5668  5774  5776  5778  5780  5782  6039
          6041  6043  6045  6050  6052  6054  6056  6058  6060  6062  6064  6069  6071  6073
          6078  6080  6085  6087  6089  6091  6093  6095  6097  6099  6101  6359  6361  6363
          6365  6370  6372  6374  6376  6378  6380  6382  6384  6389  6391  6393  6395  6397
          6399  6401  6406  6408  6413  6415  6417  6419  6421  6423  6425  6427  6429  6685
          6687  6689  6691  6696  6698  6700  6702  6704  6706  6708  6710  6715  6717  6719
          6721  6723  6725  6727  6732  6734  6739  6741  6743  6745  6747  6749  6751  6753
          6755  7013  7015  7017  7019  7024  7026  7028  7030  7032  7034  7036  7038  7043
          7045  7047  7049  7051  7053  7055  7060  7062  7067  7069  7071  7073  7075  7077
          7079  7081  7083  7341  7343  7345  7347  7352  7354  7356  7358  7360  7362  7364
          7366  7371  7373  7375  7377  7379  7381  7383  7388  7390  7395  7397  7399  7401
          7403  7405  7407  7409  7411  7633  7635  7637  7639  7644  7646  7648  7650  7652
          7654  7656  7658  7663  7665  7667  7669  7671  7673  7675  7680  7682  7687  7689
          7691  7693  7695  7697  7699  7701  7703  7924  7926  7928  7930  7935  7937  7939
          7941  7943  7945  7947  7949  7954  7956  7958  7960  7962  7964  7966  7971  7973
          7978  7980  7982  7984  7986  7988  7990  7992  7994  8214  8216  8218  8220  8225
          8227  8229  8231  8233  8235  8237  8239  8244  8246  8248  8250  8252  8254  8256
          8261  8263  8268  8270  8272  8274  8276  8278  8280  8282  8284  8503  8505  8507
          8509  8514  8516  8518  8520  8522  8524  8526  8528  8533  8535  8537  8539  8541
          8543  8545  8550  8552  8557  8559  8561  8563  8565  8567  8569  8571  8573  8758
          8760  8762  8764  8766  8768  8770  8772  8774  8776  8778  8780  8782  8784  8786
          8791  8793  8795  8797  8799  8801  8803  8805  8810  8812  8814  8819  8821  8823
          8825  8830  8832  8837  8839  8841  8843  8845  8847  8849  8851  8853  8858  8860
          8862  8864  8866  8868  8870  8872  8874
CONCAT     194   196   198   200   202   204   206   208   210   212   214   216   218   220
           222   224   226   228   230   232   234   236   238   240   242   244   246   248
           250   252   254   256   258   260   262   264   266   268   273   275   280   282
           287   289   291   296   298   300   305   307   309   314   316   318   323   325
           327   332   334   336   341   343   345   350   352   354   359   361   363   368
           370   372   377   379   381   386   388   390   395   397   399   404   406   408
           413   415   417   422   424   426   431   433   435   440   442   444   449   451
           453   458   460   462   467   469   471   476   478   480   485   487   489   494
           496   498   503   505   507   512   514   516   521   523   525   530   532   534
           539   541   543   548   550   552   557   559   561   566   568   570   575   577
           579   584   586   588   593   595   597   599   604   609   611   613   615   617
           717   719   721   723   725   727   729   731   733   735   740   742   747   749
           754   756   758   763   765   767   772   774   776   781   783   785   790   792
           794   799   801   803   808   810   812   817   822   824   826   828   830   832
           938   943   948   950   955   957   959   964   966   971   973  1062  1064  1066
          1071  1076  1081  1083  1088  1093  1095  1100  1102  1108  1110  1112  1114  1116
          1118  1120  1122  1127  1129  1330  1332  1334  1336  1341  1343  1345  1347  1349
          1351  1356  1358  1360  1365  1371  1373  1375  1377  1379  1381  1383  1385  1392
          1394  1396  1398  1400  1402  1404  1406  1408  1410  1412  1414  1416  1421  1423
          1428  1430  1432  1437  1439  1444  1446  1451  1456  1458  1586  1588  1590  1592
          1676  1678  1680  1682  1684  1686  1691  1693  1695  1697  1699  1701  1706  1708
          1710  1712  1714  1716  1718  1720  1722  1724  1726  1728  1730  1732  1734  1736
          1738  1740  1742  1744  1749  1751  1753  1755  1757  1759  1761  1763  1765  1767
          1773  1775  1777  1783  1785  1787  1789  1791  1793  1795  1797  1802  1804  1809
          1811  1813  1818  1823  1825  1975  1977  1979  1981  1984  1986  1988  1990  1992
          1994  1997  2002  2008  2010  2012  2014  2016  2018  2020  2022  2027  2029  2031
```

```
Q 1857        2033    2035    2037    2039    2041    2043    2045    2047    2049    2051    2053    2055    2057    2059
              2061    2063    2065    2070    2072    2074    2080    2082    2084    2086    2088    2090    2092    2094    SEQ 1870
              2096    2098    2100    2102    2107    2109    2269    2272    2274    2276    2278    2280    2283    2285
              2287    2289    2292    2298    2300    2302    2308    2310    2312    2314    2316    2318    2320    2322
              2327    2329    2331    2333    2335    2337    2339    2341    2343    2345    2347    2349    2351    2353
              2355    2357    2359    2361    2363    2365    2370    2372    2374    2380    2382    2384    2386    2388
              2390    2392    2394    2396    2398    2400    2402    2407    2409    2485    2487    2594    2596    2598
              2600    2602    2855    2857    2859    2861    2866    2868    2870    2872    2874    2876    2881    2883
              2885    2890    2892    2897    2899    2901    2903    2905    2907    2909    2911    2913    3165    3167
              3169    3171    3176    3178    3180    3182    3184    3186    3191    3193    3195    3200    3202    3207
              3209    3211    3213    3215    3217    3219    3221    3223    3482    3484    3486    3488    3493    3495
              3497    3499    3501    3503    3508    3510    3512    3514    3516    3518    3520    3525    3527    3532
              3534    3536    3538    3540    3542    3544    3546    3548    3805    3807    3809    3811    3816    3818
              3820    3822    3824    3826    3831    3833    3835    3837    3839    3841    3843    3848    3850    3855
              3857    3859    3861    3863    3865    3867    3869    3871    4128    4130    4132    4134    4139    4141
              4143    4145    4147    4149    4154    4156    4158    4160    4162    4164    4166    4171    4173    4178
              4180    4182    4184    4186    4188    4190    4192    4194    4452    4454    4456    4458    4463    4465
              4467    4469    4471    4473    4478    4480    4482    4484    4486    4488    4490    4495    4497    4502
              4504    4506    4508    4510    4512    4514    4516    4518    4740    4742    4744    4746    4751    4753
              4755    4757    4759    4761    4766    4768    4770    4772    4774    4776    4778    4783    4785    4790
              4792    4794    4796    4798    4800    4802    4804    4806    5027    5029    5031    5033    5038    5040
              5042    5044    5046    5048    5053    5055    5057    5059    5061    5063    5065    5070    5072    5077
              5079    5081    5083    5085    5087    5089    5091    5093    5313    5315    5317    5319    5324    5326
              5328    5330    5332    5334    5339    5341    5343    5345    5347    5349    5351    5356    5358    5363
              5365    5367    5369    5371    5373    5375    5377    5379    5602    5604    5606    5608    5613    5615
              5617    5619    5621    5623    5628    5630    5632    5634    5636    5638    5640    5645    5647    5652
              5654    5656    5658    5660    5662    5664    5666    5668    5774    5776    5778    5780    5782    6039
              6041    6043    6045    6050    6052    6054    6056    6058    6060    6062    6064    6069    6071    6073
              6078    6080    6085    6087    6089    6091    6093    6095    6097    6099    6101    6359    6361    6363
              6365    6370    6372    6374    6376    6378    6380    6382    6384    6389    6391    6393    6395    6397
              6399    6401    6406    6408    6413    6415    6417    6419    6421    6423    6425    6427    6429    6685
              6687    6689    6691    6696    6698    6700    6702    6704    6706    6708    6710    6715    6717    6719
              6721    6723    6725    6727    6732    6734    6739    6741    6743    6745    6747    6749    6751    6753
              6755    7013    7015    7017    7019    7024    7026    7028    7030    7032    7034    7036    7038    7043
              7045    7047    7049    7051    7053    7055    7060    7062    7067    7069    7071    7073    7075    7077
              7079    7081    7083    7341    7343    7345    7347    7352    7354    7356    7358    7360    7362    7364
              7366    7371    7373    7375    7377    7379    7381    7383    7388    7390    7395    7397    7399    7401
              7403    7405    7407    7409    7411    7633    7635    7637    7639    7644    7646    7648    7650    7652
              7654    7656    7658    7663    7665    7667    7669    7671    7673    7675    7680    7682    7687    7689
              7691    7693    7695    7697    7699    7701    7703    7924    7926    7928    7930    7935    7937    7939
              7941    7943    7945    7947    7949    7954    7956    7958    7960    7962    7964    7966    7971    7973
              7978    7980    7982    7984    7986    7988    7990    7992    7994    8214    8216    8218    8220    8225
              8227    8229    8231    8233    8235    8237    8239    8244    8246    8248    8250    8252    8254    8256
              8261    8263    8268    8270    8272    8274    8276    8278    8280    8282    8284    8503    8505    8507
              8509    8514    8516    8518    8520    8522    8524    8526    8528    8533    8535    8537    8539    8541
              8543    8545    8550    8552    8557    8559    8561    8563    8565    8567    8569    8571    8573    8758
              8760    8762    8764    8766    8768    8770    8772    8774    8776    8778    8780    8782    8784    8786
              8791    8793    8795    8797    8799    8801    8803    8805    8810    8812    8814    8819    8821    8823
              8825    8830    8832    8837    8839    8841    8843    8845    8847    8849    8851    8853    8858    8860
              8862    8864    8866    8868    8870    8872    8874
CTABLE         904     905     906     907     908     909     910     911    1046    1047    1048    1049    1050    1051
              1052    1053    1214    1215    1216    1217    1218    1219    1220    1221    1534    1535    1536    1537
              1538    1539    1540    1541    1935    1936    1937    1938    1939    1940    1941    1942    2218    2219
```

Q 1858

	2220	2221	2222	2223	2224	2225	2226							
FIELD	194	196	198	200	202	204	206	208	210	212	214	216	218	220
	222	224	226	228	230	232	234	236	238	240	242	244	246	248
	250	252	254	256	258	260	262	264	266	268	273	275	280	282
	287	289	291	296	298	300	305	307	309	314	316	318	323	325
	327	332	334	336	341	343	345	350	352	354	359	361	363	368
	370	372	377	379	381	386	388	390	395	397	399	404	406	408
	413	415	417	422	424	426	431	433	435	440	442	444	449	451
	453	458	460	462	467	469	471	476	478	480	485	487	489	494
	496	498	503	505	507	512	514	516	521	523	525	530	532	534
	539	541	543	548	550	552	557	559	561	566	568	570	575	577
	579	584	586	588	593	595	597	599	604	609	611	613	615	617
	717	719	721	723	725	727	729	731	733	735	740	742	747	749
	754	756	758	763	765	767	772	774	776	781	783	785	790	792
	794	799	801	803	808	810	812	817	822	824	826	828	830	832
	938	943	948	950	955	957	959	964	966	971	973	1062	1064	1066
	1071	1076	1081	1083	1088	1093	1095	1100	1102	1108	1110	1112	1114	1116
	1118	1120	1122	1127	1129	1330	1332	1334	1336	1341	1343	1345	1347	1349
	1351	1356	1358	1360	1365	1371	1373	1375	1377	1379	1381	1383	1385	1392
	1394	1396	1398	1400	1402	1404	1406	1408	1410	1412	1414	1416	1421	1423
	1428	1430	1432	1437	1439	1444	1446	1451	1456	1458	1586	1588	1590	1592
	1676	1678	1680	1682	1684	1686	1691	1693	1695	1697	1699	1701	1706	1708
	1710	1712	1714	1716	1718	1720	1722	1724	1726	1728	1730	1732	1734	1736
	1738	1740	1742	1744	1749	1751	1753	1755	1757	1759	1761	1763	1765	1767
	1773	1775	1777	1783	1785	1787	1789	1791	1793	1795	1797	1802	1804	1809
	1811	1813	1818	1823	1825	1975	1977	1979	1981	1984	1986	1988	1990	1992
	1994	1997	2002	2008	2010	2012	2014	2016	2018	2020	2022	2027	2029	2031
	2033	2035	2037	2039	2041	2043	2045	2047	2049	2051	2053	2055	2057	2059
	2061	2063	2065	2070	2072	2074	2080	2082	2084	2086	2088	2090	2092	2094
	2096	2098	2100	2102	2107	2109	2269	2272	2274	2276	2278	2280	2283	2285
	2287	2289	2292	2298	2300	2302	2308	2310	2312	2314	2316	2318	2320	2322
	2327	2329	2331	2333	2335	2337	2339	2341	2343	2345	2347	2349	2351	2353
	2355	2357	2359	2361	2363	2365	2370	2372	2374	2380	2382	2384	2386	2388
	2390	2392	2394	2396	2398	2400	2402	2407	2409	2485	2487	2594	2596	2598
	2600	2602	2855	2857	2859	2861	2866	2868	2870	2872	2874	2876	2881	2883
	2885	2890	2892	2897	2899	2901	2903	2905	2907	2909	2911	2913	3165	3167
	3169	3171	3176	3178	3180	3182	3184	3186	3191	3193	3195	3200	3202	3207
	3209	3211	3213	3215	3217	3219	3221	3223	3482	3484	3486	3488	3493	3495
	3497	3499	3501	3503	3508	3510	3512	3514	3516	3518	3520	3525	3527	3532
	3534	3536	3538	3540	3542	3544	3546	3548	3805	3807	3809	3811	3816	3818
	3820	3822	3824	3826	3831	3833	3835	3837	3839	3841	3843	3848	3850	3855
	3857	3859	3861	3863	3865	3867	3869	3871	4128	4130	4132	4134	4139	4141
	4143	4145	4147	4149	4154	4156	4158	4160	4162	4164	4166	4171	4173	4178
	4180	4182	4184	4186	4188	4190	4192	4194	4452	4454	4456	4458	4463	4465
	4467	4469	4471	4473	4478	4480	4482	4484	4486	4488	4490	4495	4497	4502
	4504	4506	4508	4510	4512	4514	4516	4518	4740	4742	4744	4746	4751	4753
	4755	4757	4759	4761	4766	4768	4770	4772	4774	4776	4778	4783	4785	4790
	4792	4794	4796	4798	4800	4802	4804	4806	5027	5029	5031	5033	5038	5040
	5042	5044	5046	5048	5053	5055	5057	5059	5061	5063	5065	5070	5072	5077
	5079	5081	5083	5085	5087	5089	5091	5093	5313	5315	5317	5319	5324	5326
	5328	5330	5332	5334	5339	5341	5343	5345	5347	5349	5351	5356	5358	5363
	5365	5367	5369	5371	5373	5375	5377	5379	5602	5604	5606	5608	5613	5615
	5617	5619	5621	5623	5628	5630	5632	5634	5636	5638	5640	5645	5647	5652

SEQ 1871

```
Q 1859  5654  5656  5658  5660  5662  5664  5666  5668  5774  5776  5778  5780  5782  6039   SEQ 1872
        6041  6043  6045  6050  6052  6054  6056  6058  6060  6062  6064  6069  6071  6073
        6078  6080  6085  6087  6089  6091  6093  6095  6097  6099  6101  6359  6361  6363
        6365  6370  6372  6374  6376  6378  6380  6382  6384  6389  6391  6393  6395  6397
        6399  6401  6406  6408  6413  6415  6417  6419  6421  6423  6425  6427  6429  6685
        6687  6689  6691  6696  6698  6700  6702  6704  6706  6708  6710  67[illegible]  6717  6719
        6721  6723  6725  6727  6732  6734  6739  6741  6743  6745  6747  6749  6751  6753
        6755  7013  7015  7017  7019  7024  7026  7028  7030  7032  7034  7036  7038  7043
        7045  7047  7049  7051  7053  7055  7060  7062  7067  7069  7071  7073  7075  7077
        7079  7081  7083  7341  7343  7345  7347  7352  7354  7356  7358  7360  7362  7364
        7366  7371  7373  7375  7377  7379  7381  7383  7388  7390  7395  7397  7399  7401
        7403  7405  7407  7409  7411  7633  7635  7637  7639  7644  7646  7648  7650  7652
        7654  7656  7658  7663  7665  7667  7669  7671  7673  7675  7680  7682  7687  7689
        7691  7693  7695  7697  7699  7701  7703  7924  7926  7928  7930  7935  7937  7939
        7941  7943  7945  7947  7949  7954  7956  7958  7960  7962  7964  7966  7971  7973
        7978  7980  7982  7984  7986  7988  7990  7992  7994  8214  8216  8218  8220  8225
        8227  8229  8231  8233  8235  8237  8239  8244  8246  8248  8250  8252  8254  8256
        8261  8263  8268  8270  8272  8274  8276  8278  8280  8282  8284  8503  8505  8507
        8509  8514  8516  8518  8520  8522  8524  8526  8528  8533  8535  8537  8539  8541
        8543  8545  8550  8552  8557  8559  8561  8563  8565  8567  8569  8571  8573  8758
        8760  8762  8764  8766  8768  8770  8772  8774  8776  8778  8780  8782  8784  8786
        8791  8793  8795  8797  8799  8801  8803  8805  8810  8812  8814  8819  8821  8823
        8825  8830  8832  8837  8839  8841  8843  8845  8847  8849  8851  8853  8858  8860
        8862  8864  8866  8868  8870  8872  8874
GO        77    79    85    91   656   658   664   670   871   873   885   886  1012  1014
        1027  1028  1181  1183  1195  1196  1235  1247  1288  1291  1293  1295  1296  1299
        1302  1304  1306  1307  1501  1503  1515  1516  1558  1581  1598  1632  1635  1637
        1639  1640  1661  1668  1868  1870  1882  1885  1887  1897  1901  1905  1908  1910
        1912  1916  1917  1953  1969  2151  2153  2165  2168  2170  2180  2184  2188  2191
        2193  219[illegible]  2199  2200  2237  2263  2442  2444  2450  2456  2528  2530  2579  2542
        2555  2556  2561  2562  2575  2581  2650  2652  2658  2673  2675  2677  2682  2686
        2693  2695  2697  2699  2703  2820  2823  2825  2827  2962  2964  2970  2985  2987
        2989  2994  2998  3005  3007  3009  3011  3015  3133  3135  3137  3276  3278  3284
        3299  3301  3303  3308  3312  3319  3321  3323  3325  3329  3450  3452  3454  3601
        3603  3609  3624  3626  3628  3633  3637  3644  3646  3648  3650  3654  3773  3775
        3777  3924  3926  3932  3947  3949  3951  3956  3960  3967  3969  3971  3973  3977
        4096  4098  4100  4246  4248  4254  4269  4271  4273  4278  4282  4289  4291  4293
        4295  4299  4420  4422  4424  4569  4571  4577  4592  4594  4596  4601  4605  4612
        4614  4616  4618  4622  4708  4710  4712  4858  4860  4866  4881  4883  4885  4890
        4894  4901  4903  4905  4907  4911  4995  4997  4999  5144  5146  5152  5167  5169
        5171  5176  5180  5187  5189  5191  5193  5197  5281  5283  5285  5431  5433  5439
        5454  5456  5458  5463  5467  5474  5476  5478  5480  5484  5570  5572  5574  5708
        5710  5718  5722  5735  5736  5741  5742  5755  5761  5831  5833  5839  5854  5856
        5858  5863  5867  5874  5876  5878  5880  5884  6004  6007  6009  6011  6154  6156
        6162  6177  6179  6181  6186  6190  6197  6199  6201  6203  6207  6327  6329  6331
        6480  6482  6488  6503  6505  6507  6512  6516  6523  6525  6527  6529  6533  6653
        6655  6657  6808  6810  6816  6831  6833  6835  6840  6844  6851  6853  6855  6557
        6861  6981  6983  6985  7136  7138  7144  7159  7161  7163  7168  7172  7179  7181
        7183  7185  7189  7309  7311  7313  7463  7465  7471  7486  7488  7490  7495  7499
        7506  7508  7510  7512  7516  7601  7603  7605  7754  7756  7762  7777  7779  7781
        7786  7790  7797  7799  7801  7803  7807  7892  7894  7896  8044  8046  8052  8067
        8069  8071  8076  8080  8087  8089  8091  8093  8097  8182  8184  8186  8333  8335
        8341  8356  8358  8360  8365  8369  8376  8378  8380  8382  8386  8471  8473  8475
        8619    8621    8627    8642    8644    8646    8648    8653    8657    8664    8666    8668    8670    8674    SEQ 1873
        8681    8684    8686    8688    8689    8693    8695    8697    8698    8734    8736    8738    8740
MFLD     194#    194     196#    196     198#    198     200#    200     202#    202     204#    204     206#    206
         208#    208     210#    210     212#    212     214#    214     216#    216     218#    218     220#    220
         222#    222     224#    224     226#    226     228#    228     230#    230     232#    232     234#    234
         236#    236     238#    238     240#    240     242#    242     244#    244     246#    246     248#    248
         250#    250     252#    252     254#    254     256#    256     258#    258     260#    260     262#    262
         264#    264     266#    266     268#    268     273#    273     275#    275     280#    280     282#    282
         287#    287     289#    289     291#    291     296#    296     298#    298     300#    300     305#    305
         307#    307     309#    309     314#    314     316#    316     318#    318     323#    323     325#    325
         327#    327     332#    332     334#    334     336#    336     341#    341     343#    343     345#    345
         350#    350     352#    352     354#    354     359#    359     361#    361     363#    363     368#    368
         370#    370     372#    372     377#    377     379#    379     381#    381     386#    386     388#    388
         390#    390     395#    395     397#    397     399#    399     404#    404     406#    406     408#    408
         413#    413     415#    415     417#    417     422#    422     424#    424     426#    426     431#    431
         433#    433     435#    435     440#    440     442#    442     444#    444     449#    449     451#    451
         453#    453     458#    458     460#    460     462#    462     467#    467     469#    469     471#    471
         476#    476     478#    478     480#    480     485#    485     487#    487     489#    489     494#    494
         496#    496     498#    498     503#    503     505#    505     507#    507     512#    512     514#    514
         516#    516     521#    521     523#    523     525#    525     530#    530     532#    532     534#    534
         539#    539     541#    541     543#    543     548#    548     550#    550     552#    552     557#    557
         559#    559     561#    561     566#    566     568#    568     570#    570     575#    575     577#    577
         579#    579     584#    584     586#    586     588#    588     593#    593     595#    595     597#    597
         599#    599     604#    604     609#    609     611#    611     613#    613     615#    615     617#    617
         717#    717     719#    719     721#    721     723#    723     725#    725     727#    727     729#    729
         731#    731     733#    733     735#    735     740#    740     742#    742     747#    747     749#    749
         754#    754     756#    756     758#    758     763#    763     765#    765     767#    767     772#    772
         774#    774     776#    776     781#    781     783#    783     785#    785     790#    790     792#    792
         794#    794     799#    799     801#    801     803#    803     808#    808     810#    810     812#    812
         817#    817     822#    822     824#    824     826#    826     828#    828     830#    830     832#    832
         938#    938     943#    943     948#    948     950#    950     955#    955     957#    957     959#    959
         964#    964     966#    966     971#    971     973#    973    1062#   1062    1064#   1064    1066#   1066
        1071#   1071    1076#   1076    1081#   1081    1083#   1083    1088#   1088    1093#   1093    1095#   1095
        1100#   1100    1102#   1102    1108#   1108    1110#   1110    1112#   1112    1114#   1114    1116#   1116
        1118#   1118    1120#   1120    1122#   1122    1127#   1127    1129#   1129    1330#   1330    1332#   1332
        1334#   1334    1336#   1336    1341#   1341    1343#   1343    1345#   1345    1347#   1347    1349#   1349
        1351#   1351    1356#   1356    1358#   1358    1360#   1360    1365#   1365    1371#   1371    1373#   1373
        1375#   1375    1377#   1377    1379#   1379    1381#   1381    1383#   1383    1385#   1385    1392#   1392
        1394#   1394    1396#   1396    1398#   1398    1400#   1400    1402#   1402    1404#   1404    1406#   1406
        1408#   1408    1410#   1410    1412#   1412    1414#   1414    1416#   1416    1421#   1421    1423#   1423
        1428#   1428    1430#   1430    1432#   1432    1437#   1437    1439#   1439    1444#   1444    1446#   1446
        1451#   1451    1456#   1456    1458#   1458    1586#   1586    1588#   1588    1590#   1590    1592#   1592
        1676#   1676    1678#   1678    1680#   1680    1682#   1682    1684#   1684    1686#   1686    1691#   1691
        1693#   1693    1695#   1695    1697#   1697    1699#   1699    1701#   1701    1706#   1706    1708#   1708
        1710#   1710    1712#   1712    1714#   1714    1716#   1716    1718#   1718    1720#   1720    1722#   1722
        1724#   1724    1726#   1726    1728#   1728    1730#   1730    1732#   1732    1734#   1734    1736#   1736
        1738#   1738    1740#   1740    1742#   1742    1744#   1744    1749#   1749    1751#   1751    1753#   1753
        1755#   1755    1757#   1757    1759#   1759    1761#   1761    1763#   1763    1765#   1765    1767#   1767
        1773#   1773    1775#   1775    1777#   1777    1783#   1783    1785#   1785    1787#   1787    1789#   1789
        1791#   1791    1793#   1793    1795#   1795    1797#   1797    1802#   1802    1804#   1804    1809#   1809
        1811#   1811    1813#   1813    1818#   1818    1823#   1823    1825#   1825    1975#   1975    1977#   1977
        1979#   1979    1981#   1981    1984#   1984    1986#   1986    1988#   1988    1990#   1990    1992#   1992
        1994#   1994    1997#   1997    2002#   2002    2008#   2008    2010#   2010    2012#   2012    2014#   2014
```

```
DFPTAA
DFPTAA

  322
  323
  324
  325
  326
  327
  328
  329
  330
  331
  332
  333
  334
  335
  336
  337
  338
  339
  340
  341
  342
  343
  344
  345
  346
  347
  348
  349
  350
  351
  352
  353
  354
  355
  356
  357
  358
  359
  360
  361
  362
  363
  364
  365
  366
  367
  368
  369
  370
  371
  372
  373
  374
  375
  376
```

```
2016#  2016   2018#  2018   2020#  2020   2022#  2022   2027#  2027   2029#  2029   2031#  2031
2033#  2033   2035#  2035   2037#  2037   2039#  2039   2041#  2041   2043#  2043   2045#  2045      SEQ 1874
2047#  2047   2049#  2049   2051#  2051   2053#  2053   2055#  2055   2057#  2057   2059#  2059
2061#  2061   2063#  2063   2065#  2065   2070#  2070   2072#  2072   2074#  2074   2080#  2080
2082#  2082   2084#  2084   2086#  2086   2088#  2088   2090#  2090   2092#  2092   2094#  2094
2096#  2096   2098#  2098   2100#  2100   2102#  2102   2107#  2107   2109#  2109   2269#  2269
2272#  2272   2274#  2274   2276#  2276   2278#  2278   2280#  2280   2283#  2283   2285#  2285
2287#  2287   2289#  2289   2292#  2292   2298#  2298   2300#  2300   2302#  2302   2308#  2308
2310#  2310   2312#  2312   2314#  2314   2316#  2316   2318#  2318   2320#  2320   2322#  2322
2327#  2327   2329#  2329   2331#  2331   2333#  2333   2335#  2335   2337#  2337   2339#  2339
2341#  2341   2343#  2343   2345#  2345   2347#  2347   2349#  2349   2351#  2351   2353#  2353
2355#  2355   2357#  2357   2359#  2359   2361#  2361   2363#  2363   2365#  2365   2370#  2370
2372#  2372   2374#  2374   2380#  2380   2382#  2382   2384#  2384   2386#  2386   2388#  2388
2390#  2390   2392#  2392   2394#  2394   2396#  2396   2398#  2398   2400#  2400   2402#  2402
2407#  2407   2409#  2409   2485#  2485   2487#  2487   2594#  2594   2596#  2596   2598#  2598
2600#  2600   2602#  2602   2855#  2855   2857#  2857   2859#  2859   2861#  2861   2866#  2866
2868#  2868   2870#  2870   2872#  2872   2874#  2874   2876#  2876   2881#  2881   2883#  2883
2885#  2885   2890#  2890   2892#  2892   2897#  2897   2899#  2899   2901#  2901   2903#  2903
2905#  2905   2907#  2907   2909#  2909   2911#  2911   2913#  2913   3165#  3165   3167#  3167
3169#  3169   3171#  3171   3176#  3176   3178#  3178   3180#  3180   3182#  3182   3184#  3184
3186#  3186   3191#  3191   3193#  3193   3195#  3195   3200#  3200   3202#  3202   3207#  3207
3209#  3209   3211#  3211   3213#  3213   3215#  3215   3217#  3217   3219#  3219   3221#  3221
3223#  3223   3482#  3482   3484#  3484   3486#  3486   3488#  3488   3493#  3493   3495#  3495
3497#  3497   3499#  3499   3501#  3501   3503#  3503   3508#  3508   3510#  3510   3512#  3512
3514#  3514   3516#  3516   3518#  3518   3520#  3520   3525#  3525   3527#  3527   3532#  3532
3534#  3534   3536#  3536   3538#  3538   3540#  3540   3542#  3542   3544#  3544   3546#  3546
3548#  3548   3805#  3805   3807#  3807   3809#  3809   3811#  3811   3816#  3816   3818#  3818
3820#  3820   3822#  3822   3824#  3824   3826#  3826   3831#  3831   3833#  3833   3835#  3835
3837#  3837   3839#  3839   3841#  3841   3843#  3843   3848#  3848   3850#  3850   3855#  3855
3857#  3857   3859#  3859   3861#  3861   3863#  3863   3865#  3865   3867#  3867   3869#  3869
3871#  3871   4128#  4128   4130#  4130   4132#  4132   4134#  4134   4139#  4139   4141#  4141
4143#  4143   4145#  4145   4147#  4147   4149#  4149   4154#  4154   4156#  4156   4158#  4158
4160#  4160   4162#  4162   4164#  4164   4166#  4166   4171#  4171   4173#  4173   4178#  4178
4180#  4180   4182#  4182   4184#  4184   4186#  4186   4188#  4188   4190#  4190   4192#  4192
4194#  4194   4452#  4452   4454#  4454   4456#  4456   4458#  4458   4463#  4463   4465#  4465
4467#  4467   4469#  4469   4471#  4471   4473#  4473   4478#  4478   4480#  4480   4482#  4482
4484#  4484   4486#  4486   4488#  4488   4490#  4490   4495#  4495   4497#  4497   4502#  4502
4504#  4504   4506#  4506   4508#  4508   4510#  4510   4512#  4512   4514#  4514   4516#  4516
4518#  4518   5740#  4740   4742#  4742   4744#  4744   4746#  4746   4751#  4751   4753#  4753
4755#  4755   4757#  4757   4759#  4759   4761#  4761   4766#  4766   4768#  4768   4770#  4770
4772#  4772   4774#  4774   4776#  4776   4778#  4778   4783#  4783   4785#  4785   4790#  4790
4792#  4792   4794#  4794   4796#  4796   4798#  4798   4800#  4800   4802#  4802   4804#  4804
4806#  4806   5027#  5027   5029#  5029   5031#  5031   5033#  5033   5038#  5038   5040#  5040
5042#  5042   5044#  5044   5046#  5046   5048#  5048   5053#  5053   5055#  5055   5057#  5057
5059#  5059   5061#  5061   5063#  5063   5065#  5065   5070#  5070   5072#  5072   5077#  5077
5079#  5079   5081#  5081   5083#  5083   5085#  5085   5087#  5087   5089#  5089   5091#  5091
5093#  5093   5313#  5313   5315#  5315   5317#  5317   5319#  5319   5324#  5324   5326#  5326
5328#  5328   5330#  5330   5332#  5332   5334#  5334   5339#  5339   5341#  5341   5343#  5343
5345#  5345   5347#  5347   5349#  5349   5351#  5351   5356#  5356   5358#  5358   5363#  5363
5365#  5365   5367#  5367   5369#  5369   5371#  5371   5373#  5373   5375#  5375   5377#  5377
5379#  5379   5602#  5602   5604#  5604   5606#  5606   5608#  5608   5613#  5613   5615#  5615
5617#  5617   5619#  5619   5621#  5621   5623#  5623   5628#  5628   5630#  5630   5632#  5632
5634#  5634   5636#  5636   5638#  5638   5640#  5640   5645#  5645   5647#  5647   5652#  5652
```

```
         5654#  5654  5656#  5656  5658#  5658  5660#  5660  5662#  5662  5664#  5664  5666#  5666     SEQ 1875
         5668#  5668  5774#  5774  5776#  5776  5778#  5778  5780#  5780  5782#  5782  6039#  6039
         6041#  6041  6043#  6043  6045#  6045  6050#  6050  6052#  6052  6054#  6054  6056#  6056
         6058#  6058  6060#  6060  6062#  6062  6064#  6064  6069#  6069  6071#  6071  6073#  6073
         6078#  6078  6080#  6080  6085#  6085  6087#  6087  6089#  6089  6091#  6091  6093#  6093
         6095#  6095  6097#  6097  6099#  6099  6101#  6101  6359#  6359  6361#  6361  6363#  6363
         6365#  6365  6370#  6370  6372#  6372  6374#  6374  6376#  6376  6378#  6378  6380#  6380
         6382#  6382  6384#  6384  6389#  6389  6391#  6391  6393#  6393  6395#  6395  6397#  6397
         6399#  6399  6401#  6401  6406#  6406  6408#  6408  6413#  6413  6415#  6415  6417#  6417
         6419#  6419  6421#  6421  6423#  6423  6425#  6425  6427#  6427  6429#  6429  6685#  6685
         6687#  6687  6689#  6689  6691#  6691  6696#  6696  6698#  6698  6700#  6700  6702#  6702
         6704#  6704  6706#  6706  6708#  6708  6710#  6710  6715#  6715  6717#  6717  6719#  6719
         6721#  6721  6723#  6723  6725#  6725  6727#  6727  6732#  6732  6734#  6734  6739#  6739
         6741#  6741  6743#  6743  6745#  6745  6747#  6747  6749#  6749  6751#  6751  6753#  6753
         6755#  6755  7013#  7013  7015#  7015  7017#  7017  7019#  7019  7024#  7024  7026#  7026
         7028#  7028  7030#  7030  7032#  7032  7034#  7034  7036#  7036  7038#  7038  7043#  7043
         7045#  7045  7047#  7047  7049#  7049  7051#  7051  7053#  7053  7055#  7055  7060#  7060
         7062#  7062  7067#  7067  7069#  7069  7071#  7071  7073#  7073  7075#  7075  7077#  7077
         7079#  7079  7081#  7081  7083#  7083  7341#  7341  7343#  7343  7345#  7345  7347#  7347
         7352#  7352  7354#  7354  7356#  7356  7358#  7358  7360#  7360  7362#  7362  7364#  7364
         7366#  7366  7371#  7371  7373#  7373  7375#  7375  7377#  7377  7379#  7379  7381#  7381
         7383#  7383  7388#  7388  7390#  7390  7395#  7395  7397#  7397  7399#  7399  7401#  7401
         7403#  7403  7405#  7405  7407#  7407  7409#  7409  7411#  7411  7633#  7633  7635#  7635
         7637#  7637  7639#  7639  7644#  7644  7646#  7646  7648#  7648  7650#  7650  7652#  7652
         7654#  7654  7656#  7656  7658#  7658  7663#  7663  7665#  7665  7667#  7667  7669#  7669
         7671#  7671  7673#  7673  7675#  7675  7680#  7680  7682#  7682  7687#  7687  7689#  7689
         7691#  7691  7693#  7693  7695#  7695  7697#  7697  7699#  7699  7701#  7701  7703#  7703
         7924#  7924  7926#  7926  7928#  7928  7930#  7930  7935#  7935  7937#  7937  7939#  7939
         7941#  7941  7943#  7943  7945#  7945  7947#  7947  7949#  7949  7954#  7954  7956#  7956
         7958#  7958  7960#  7960  7962#  7962  7964#  7964  7966#  7966  7971#  7971  7973#  7973
         7978#  7978  7980#  7980  7982#  7982  7984#  7984  7986#  7986  7988#  7988  7990#  7990
         7992#  7992  7994#  7994  8214#  8214  8216#  8216  8218#  8218  8220#  8220  8225#  8225
         8227#  8227  8229#  8229  8231#  8231  8233#  8233  8235#  8235  8237#  8237  8239#  8239
         8244#  8244  8246#  8246  8248#  8248  8250#  8250  8252#  8252  8254#  8254  8256#  8256
         8261#  8261  8263#  8263  8268#  8268  8270#  8270  8272#  8272  8274#  8274  8276#  8276
         8278#  8278  8280#  8280  8282#  8282  8284#  8284  8503#  8503  8505#  8505  8507#  8507
         8509#  8509  8514#  8514  8516#  8516  8518#  8518  8520#  8520  8522#  8522  8524#  8524
         8526#  8526  8528#  8528  8533#  8533  8535#  8535  8537#  8537  8539#  8539  8541#  8541
         8543#  8543  8545#  8545  8550#  8550  8552#  8552  8557#  8557  8559#  8559  8561#  8561
         8563#  8563  8565#  8565  8567#  8567  8569#  8569  8571#  8571  8573#  8573  8758#  8758
         8760#  8760  8762#  8762  8764#  8764  8766#  8766  8768#  8768  8770#  8770  8772#  8772
         8774#  8774  8776#  8776  8778#  8778  8780#  8780  8782#  8782  8784#  8784  8786#  8786
         8791#  8791  8793#  8793  8795#  8795  8797#  8797  8799#  8799  8801#  8801  8803#  8803
         8805#  8805  8810#  8810  8812#  8812  8814#  8814  8819#  8819  8821#  8821  8823#  8823
         8825#  8825  8830#  8830  8832#  8832  8837#  8837  8839#  8839  8841#  8841  8843#  8843
         8845#  8845  8847#  8847  8849#  8849  8851#  8851  8853#  8853  8858#  8858  8860#  8860
         8862#  8862  8864#  8864  8866#  8866  8868#  8868  8870#  8870  8872#  8872  8874#  8874
MWORD     194    196    198    200    202    204    206    208    210    212    214    216    218    220
          222    224    226    228    230    232    234    236    238    240    242    244    246    248
          250    252    254    256    258    260    262    264    266    268    273    275    280    282
          287    289    291    296    298    300    305    307    309    314    316    318    323    325
          327    332    334    336    341    343    345    350    352    354    359    361    363    368
          370    372    377    379    381    386    388    390    395    397    399    404    406    408
```

```
 413   415   417   422   424   426   431   433   435   440   442   444   449   451      SEQ 1876
 453   458   460   462   467   469   471   476   478   480   485   487   489   494
 496   498   503   505   507   512   514   516   521   523   525   530   532   534
 539   541   543   548   550   552   557   559   561   566   568   570   575   577
 579   584   586   588   593   595   597   599   604   609   611   613   615   617
 717   719   721   723   725   727   729   731   733   735   740   742   747   749
 754   756   758   763   765   767   772   774   776   781   783   785   790   792
 794   799   801   803   808   810   812   817   822   824   826   828   830   832
 938   943   948   950   955   957   959   964   966   971   973  1062  1064  1066
1071  1076  1081  1083  1088  1093  1095  1100  1102  1108  1110  1112  1114  1116
1118  1120  1122  1127  1129  1330  1332  1334  1336  1341  1343  1345  1347  1349
1351  1356  1358  1360  1365  1371  1373  1375  1377  1379  1381  1383  1385  1392
1394  1396  1398  1400  1402  1404  1406  1408  1410  1412  1414  1416  1421  1423
1428  1430  1432  1437  1439  1444  1446  1451  1456  1458  1586  1588  1590  1592
1676  1678  1680  1682  1684  1686  1691  1693  1695  1697  1699  1701  1706  1708
1710  1712  1714  1716  1718  1720  1722  1724  1726  1728  1730  1732  1734  1736
1738  1740  1742  1744  1749  1751  1753  1755  1757  1759  1761  1763  1765  1767
1773  1775  1777  1783  1785  1787  1789  1791  1793  1795  1797  1802  1804  1809
1811  1813  1818  1823  1825  1975  1977  1979  1981  1984  1986  1988  1990  1992
1994  1997  2002  2008  2010  2012  2014  2016  2018  2020  2022  2027  2029  2031
2033  2035  2037  2039  2041  2043  2045  2047  2049  2051  2053  2055  2057  2059
2061  2063  2065  2070  2072  2074  2080  2082  2084  2086  2088  2090  2092  2094
2096  2098  2100  2102  2107  2109  2269  2272  2274  2276  2278  2280  2283  2285
2287  2289  2292  2298  2300  2302  2308  2310  2312  2314  2316  2318  2320  2322
2327  2329  2331  2333  2335  2337  2339  2341  2343  2345  2347  2349  2351  2353
2355  2357  2359  2361  2363  2365  2370  2372  2374  2380  2382  2384  2386  2388
2390  2392  2394  2396  2398  2400  2402  2407  2409  2485  2487  2594  2596  2598
2600  2602  2855  2857  2859  2861  2866  2868  2870  2872  2874  2876  2881  2883
2885  2890  2892  2897  2899  2901  2903  2905  2907  2909  2911  2913  3165  3167
3169  3171  3176  3178  3180  3182  3184  3186  3191  3193  3195  3200  3202  3207
3209  3211  3213  3215  3217  3219  3221  3223  3482  3484  3486  3488  3493  3495
3497  3499  3501  3503  3508  3510  3512  3514  3516  3518  3520  3525  3527  3532
3534  3536  3538  3540  3542  3544  3546  3548  3805  3807  3809  3811  3816  3818
3820  3822  3824  3826  3831  3833  3835  3837  3839  3841  3843  3848  3850  3855
3857  3859  3861  3863  3865  3867  3869  3871  4128  4130  4132  4134  4139  4141
4143  4145  4147  4149  4154  4156  4158  4160  4162  4164  4166  4171  4173  4178
4180  4182  4184  4186  4188  4190  4192  4194  4452  4454  4456  4458  4463  4465
4467  4469  4471  4473  4478  4480  4482  4484  4486  4488  4490  4495  4497  4502
4504  4506  4508  4510  4512  4514  4516  4518  4740  4742  4744  4746  4751  4753
4755  4757  4759  4761  4766  4768  4770  4772  4774  4776  4778  4783  4785  4790
4792  4794  4796  4798  4800  4802  4804  4806  5027  5029  5031  5033  5038  5040
5042  5044  5046  5048  5053  5055  5057  5059  5061  5063  5065  5070  5072  5077
5079  5081  5083  5085  5087  5089  5091  5093  5313  5315  5317  5319  5324  5326
5328  5330  5332  5334  5339  5341  5343  5345  5347  5349  5351  5356  5358  5363
5365  5367  5369  5371  5373  5375  5377  5379  5602  5604  5606  5608  5613  5615
5617  5619  5621  5623  5628  5630  5632  5634  5636  5638  5640  5645  5647  5652
5654  5656  5658  5660  5662  5664  5666  5668  5774  5776  5778  5780  5782  6039
6041  6043  6045  6050  6052  6054  6056  6058  6060  6062  6064  6069  6071  6073
6078  6080  6085  6087  6089  6091  6093  6095  6097  6099  6101  6359  6361  6363
6365  6370  6372  6374  6376  6378  6380  6382  6384  6389  6391  6393  6395  6397
6399  6401  6406  6408  6413  6415  6417  6419  6421  6423  6425  6427  6429  6685
6687  6689  6691  6696  6698  6700  6702  6704  6706  6708  6710  6715  6717  6719
6721  6723  6725  6727  6732  6734  6739  6741  6743  6745  6747  6749  6751  6753
```

```
          6755  7013  7015  7017  7019  7024  7026  7028  7030  7032  7034  7036  7038  7043    SEQ 1877
          7045  7047  7049  7051  7053  7055  7060  7062  7067  7069  7071  7073  7075  7077
          7079  7081  7083  7341  7343  7345  7347  7352  7354  7356  7358  7360  7362  7364
          7366  7371  7373  7375  7377  7379  7381  7383  7388  7390  7395  7397  7399  7401
          7403  7405  7407  7409  7411  7633  7635  7637  7639  7644  7646  7648  7650  7652
          7654  7656  7658  7663  7665  7667  7669  7671  7673  7675  7680  7682  7687  7689
          7691  7693  7695  7697  7699  7701  7703  7924  7926  7928  7930  7935  7937  7939
          7941  7943  7945  7947  7949  7954  7956  7958  7960  7962  7964  7966  7971  7973
          7978  7980  7982  7984  7986  7988  7990  7992  7994  8214  8216  8218  8220  8225
          8227  8229  8231  8233  8235  8237  8239  8244  8246  8248  8250  8252  8254  8256
          8261  8263  8268  8270  8272  8274  8276  8278  8280  8282  8284  8503  8505  8507
          8509  8514  8516  8518  8520  8522  8524  8526  8528  8533  8535  8537  8539  8541
          8543  8545  8550  8552  8557  8559  8561  8563  8565  8567  8569  8571  8573  8758
          8760  8762  8764  8766  8768  8770  8772  8774  8776  8778  8780  8782  8784  8786
          8791  8793  8795  8797  8799  8801  8803  8805  8810  8812  8814  8819  8821  8823
          8825  8830  8832  8837  8839  8841  8843  8845  8847  8849  8851  8853  8858  8860
          8862  8864  8866  8868  8870  8872  8874
PDEC      1273  1276  1623  1626
PNTDEC    1282  1318  2849  3159  3476  3799  4122  4446  4734  5021  5307  5596  6033  6353
          6679  7007  7335  7627  7918  8208  8497
PNTHW     1322  1325  1651  1654  2589  2846  3156  3473  3796  4119  4443  4731  5018  5304
          5593  5769  6030  6350  6676  7004  7332  7624  7915  8205  8494  8751
PNTMSG    1271  1272  1274  1275  1277  1279  1283  1315  1319  1320  1323  1621  1622  1624
          1625  1627  1648  1649  1652  2587  2843  2847  2850  3153  3157  3160  3470  3474
          3477  3793  3797  3800  4116  4120  4123  4440  4444  4447  4728  4732  4735  5015
          5019  5022  5301  5305  5308  5590  5594  5597  5767  6027  6031  6034  6347  6351
          6354  6673  6677  6680  7001  7005  7008  7329  7333  7336  7621  7625  7628  7912
          7916  7919  8202  8206  8209  8491  8495  8498  8748  8752  8753
RTN         80   105   659   684   874   900   916   923   929  1015  1042  1184  1210  1230
          1243  1256  1314  1326  1504  1530  1554  1584  1607  1655  1665  1670  1871  1931
          1948  1956  1965  1971  2154  2214  2232  2240  2250  2259  2265  2445  2470  2531
          2568  2582  2590  2653  2717  2801  2828  2834  2851  2965  3029  3113  3138  3144
          3161  3279  3343  3427  3455  3461  3478  3604  3668  3752  3778  3784  3801  3927
          3991  4075  4101  4107  4124  4249  4313  4397  4425  4431  4448  4572  4636  4685
          4713  4719  4736  4861  4925  4974  5000  5006  5023  5147  5211  5260  5286  5292
          5309  5434  5498  5547  5575  5581  5598  5711  5748  5762  5770  5834  5898  5982
          6012  6018  6035  6157  6221  6305  6332  6338  6355  6483  6547  6631  6658  6664
          6681  6811  6875  6959  6986  6992  7009  7139  7203  7287  7314  7320  7337  7466
          7530  7579  7606  7612  7629  7757  7821  7870  7897  7903  7920  8047  8111  8160
          8187  8193  8210  8336  8400  8449  8476  8482  8499  8622  8710  8724  8741  8754
SCOPER      98   677   893  1035  1203  1523  1924  2207  2463  2546  2564  2710  3022  3336
          3661  3984  4306  4629  4918  5204  5491  5726  5744  5891  6214  6540  6868  7196
          7523  7814  8104  8393  8703
TMSG      1274  1277  1283  1319  1624  1627  2847  2850  3157  3160  3474  3477  3797  3800
          4120  4123  4444  4447  4732  4735  5019  5022  5305  5308  5594  5597  6031  6034
          6351  6354  6677  6680  7005  7008  7333  7336  7625  7628  7916  7919  8206  8209
          8495  8498  8752
TMSGC     1271  1272  1275  1279  1315  1320  1323  1621  1622  1625  1648  1649  1652  2587
          2843  3153  3470  3793  4116  4440  4728  5015  5301  5590  5767  6027  6347  6673
          7001  7329  7621  7912  8202  8491  8748  8753
TTABLE    2721  2722  2726  2727  2728  2729  2733  2734  2735  2736  2740  2741  2742  2743
          2747  2748  2749  2750  2755  2756  2757  2758  2762  2763  2764  2765  2769  2770
          2771  2772  2776  2777  2778  2779  2783  2784  2785  2786  2790  2791  2792  2793
```

```
2794  2795  3033  3034  3038  3039  3040  3041  3045  3046  3047  3048  3052  3053    SEQ 1878
3054  3055  3059  3060  3061  3062  3067  3068  3069  3070  3074  3075  3076  3077
3081  3082  3083  3084  3088  3089  3090  3091  3095  3096  3097  3098  3102  3103
3104  3105  3106  3107  3347  3348  3352  3353  3354  3355  3359  3360  3361  3362
3366  3367  3368  3369  3373  3374  3375  3376  3381  3382  3383  3384  3388  3389
3390  3391  3395  3396  3397  3398  3402  3403  3404  3405  3409  3410  3411  3412
3416  3417  3418  3419  3420  3421  3672  3673  3677  3678  3679  3680  3684  3685
3686  3687  3691  3692  3693  3694  3698  3699  3700  3701  3706  3707  3708  3709
3713  3714  3715  3716  3720  3721  3722  3723  3727  3728  3729  3730  3734  3735
3736  3737  3741  3742  3743  3744  3745  3746  3995  3996  4000  4001  4002  4003
4007  4008  4009  4010  4014  4015  4016  4017  4021  4022  4023  4024  4029  4030
4031  4032  4036  4037  4038  4039  4043  4044  4045  4046  4050  4051  4052  4053
4057  4058  4059  4060  4064  4065  4066  4067  4068  4069  4317  4318  4322  4323
4324  4325  4329  4330  4331  4332  4336  4337  4338  4339  4343  4344  4345  4346
4351  4352  4353  4354  4358  4359  4360  4361  4365  4366  4367  4368  4372  4373
4374  4375  4379  4380  4381  4382  4386  4387  4388  4389  4390  4391  4640  4641
4645  4646  4647  4648  4652  4653  4654  4655  4660  4661  4662  4663  4667  4668
4669  4670  4674  4675  4676  4677  4678  4679  4929  4930  4934  4935  4936  4937
4941  4942  4943  4944  4948  4949  4950  4951  4956  4957  4958  4959  4963  4964
4965  4966  4967  4968  5215  5216  5220  5221  5222  5223  5227  5228  5229  5230
5234  5235  5236  5237  5241  5242  5243  5244  5249  5250  5251  5252  5253  5254
5502  5503  5507  5508  5509  5510  5514  5515  5516  5517  5521  5522  5523  5524
5528  5529  5530  5531  5536  5537  5538  5539  5540  5541  5902  5903  5907  5908
5909  5910  5914  5915  5916  5917  5921  5922  5923  5924  5928  5929  5930  5931
5936  5937  5938  5939  5943  5944  5945  5946  5950  5951  5952  5953  5957  5958
5959  5960  5964  5965  5966  5967  5971  5972  5973  5974  5975  5976  6225  6226
6230  6231  6232  6233  6237  6238  6239  6240  6244  6245  6246  6247  6251  6252
6253  6254  6258  6259  6260  6261  6266  6267  6268  6269  6273  6274  6275  6276
6280  6281  6282  6283  6287  6288  6289  6290  6294  6295  6296  6297  6298  6299
6551  6552  6556  6557  6558  6559  6563  6564  6565  6566  6570  6571  6572  6573
6577  6578  6579  6580  6585  6586  6587  6588  6592  6593  6594  6595  6599  6600
6601  6602  6606  6607  6608  6609  6613  6614  6615  6616  6620  6621  6622  6623
6624  6625  6879  6880  6884  6885  6886  6887  6891  6892  6893  6894  6898  6899
6900  6901  6905  6906  6907  6908  6913  6914  6915  6916  6920  6921  6922  6923
6927  6928  6929  6930  6934  6935  6936  6937  6941  6942  6943  6944  6948  6949
6950  6951  6952  6953  7207  7208  7212  7213  7214  7215  7219  7220  7221  7222
7226  7227  7228  7229  7233  7234  7235  7236  7240  7241  7242  7243  7248  7249
7250  7251  7255  7256  7257  7258  7262  7263  7264  7265  7269  7270  7271  7272
7276  7277  7278  7279  7280  7281  7534  7535  7539  7540  7541  7542  7546  7547
7548  7549  7553  7554  7555  7556  7560  7561  7562  7563  7568  7569  7570  7571
7572  7573  7825  7826  7830  7831  7832  7833  7837  7838  7839  7840  7844  7845
7846  7847  7851  7852  7853  7854  7858  7859  7860  7861  7862  7863  8115  8116
8120  8121  8122  8123  8127  8128  8129  8130  8134  8135  8136  8137  8141  8142
8143  8144  8149  8150  8151  8152  8153  8154  8404  8405  8409  8410  8411  8412
8416  8417  8418  8419  8423  8424  8425  8426  8430  8431  8432  8433  8438  8439
8440  8441  8442  8443  8714  8715  8716  8717  8718  8719  8720
```

```
                                      TITLE   DFPTAA
                                      SUBTTL  CBUS MODULE TESTS

                                      SALL

                                      SEARCH  DFPTA,MONSYM

                              ; INTERN's

                                      ENTRY   TSTC40,TSTC41,TSTC42,TSTC43,TSTC44,TSTC45,TSTC46,TSTC47,TSTC50
                                      ENTRY   TSTC51,TSTC52,TSTC53,TSTC54,TSTC55,TSTC56,TSTC57,TSTC60

                              ; EXTERN's located in DFPTA1.MAC (Mainline code..)

                                      EXTERN  UDEBUG,BUFF,CBUF,TLOAD,TRACE,TSTFLG,TSTSUB,ODELAY
                                      EXTERN  BEXEC,BBPNT,TEXEC,TTPNT,CEXEC,CCPNT,CSTATF,CEBUFA

                              ; EXTERN's located in DFPTAI.MAC (Port Device Handling Routines module)

                                      EXTERN  LDRAR,LDCSR,RDEBUF,IPACLR,CHINIT,GENCCW,GETLOG,LOGBUF

                              ; EXTERN's located in DFPTAU.MAC (Utility Routines module)

                                      EXTERN  PATBUF,BUFCOM,SCOSW


                              ;#***********************************************************************
                              ; ZA - Address for use in DDT
                              ;#***********************************************************************

     000000'                  ZA:                                     ; address of 00000'
```

```
                                         ;#*********************************************************************
                                         ;* Test 40 - CMUX Selected by PLINTOCMUX
                                         ;
                                         ; Description:  This test verifies the  PLINTOCMUX  select  line.
                                         ;               First the PLIN is loaded with 377 and the FMTR is
                                         ;               loaded with 777777,,777400. Then the PLIN is read
                                         ;               and verified and the formatter read and verified.
                                         ;
                                         ; Procedure:    KL> Port Clear
                                         ;               KL> Load test microcode
                                         ;
                                         ;               KL> Start microcode at location 0
                                         ;               UC> Load PLIN with 377
                                         ;               UC> Load FMTR with 777777,,777400
                                         ;               UC> Read PLIN through CMUX with PLINTOCMUX
                                         ;                   signal asserted
                                         ;               UC> Verify data is 000000,,000377
                                         ;               UC> if passed - set bit 12 (DISABLE Complete)
                                         ;               UC> if failed - set bit 13 (ENABLE Complete)
                                         ;
                                         ;               KL> Start microcode at location 100
                                         ;               UC> Load PLIN with 377
                                         ;               UC> Load FMTR with 777777,,777400
                                         ;               UC> Read PLIN through CMUX with PLINTOCMUX
                                         ;                   signal not asserted
                                         ;               UC> Verify data is 777777,,777400
                                         ;               UC> if passed - set bit 12 (DISABLE Complete)
                                         ;               UC> if failed - set bit 13 (ENABLE Complete)
                                         ;
                                         ; Failure:      ---
                                         ;#*********************************************************************

                                         ; Test data

        000000' 254 00 0 00 000011'      TSTC40: JRST    TG40                    ; go start test
        000001' 100604  000040                   CBUS!NDMP!NDCB!ZCBUS!40         ; test mask
        000002' 000052' 004735'                  T40M,,[ASCIZ ^CMUX Selected by PLINTOCMUX^]
        000003' 004743' 000000                   [EXP C8,MLAST!C23],,0
        000004' 000000  000163'                  TSTC41                          ; failure test table
        000005' 000000  000413'                  TSTC42                          ;   ...
        000006' 000000  000503'                  TSTC43
        000007' 000000  000572'                  TSTC44
        000010' 777777  777777                   -1

                                         ; Start test

        000011' 201 00 0 00 000000'      TG40:   MOVEI   ZA                      ; get address of module start
        000012' 260 17 0 00 000000*              GO      TRACE                   ; handle trace output
        000013' 201 01 0 00 000052'              MOVEI   1,T40M                  ; set up microcode address
        000014' 260 17 0 00 000000*              GO      TLOAD                   ; load/verify it
        000015' 263 17 0 00 000000               RTN                             ; failed - exit test
```

```

                                        ; Initialization

000016' 400 15 0 00 000000      TL40:   SETZ    ERFLG,                  ; clear error flag
000017' 260 17 0 00 000000*             GO      IPACLR                  ; clear port
000020' 402 00 0 00 000000*             SETZM   TSTSUB                  ; initialize subtest number
000021' 201 06 0 00 000034'             MOVEI   6,TS40                  ; get sstep table address

                                        ; Loop on test execute table entries

000022' 260 17 0 00 000017*     TA40:   GO      IPACLR                  ; clear port
000023' 260 17 0 00 000000*     TB40:   GO      TEXEC                   ; execute table entry
000024' 254 00 0 00 000033'              JRST   TX40                    ; end of sstep table
000025' 254 00 0 00 000023'              JRST   TB40                    ; keep looping after call
000026' 474 15 0 00 000000               SETO   ERFLG,                  ; error occurred

                                        ; Handle error printouts and scope looping

000027' 027 00 0 00 000043'             SCOPER  MA40                    ; print error message
000030' 254 00 0 00 000016'             JRST    TL40                    ; loop on error
000031' 254 00 0 00 000033'             JRST    TX40                    ; altmode exit
000032' 322 15 0 00 000022'             JUMPE   ERFLG,TA40              ; do next sstep table entry

                                        ; End of test

000033' 263 17 0 00 000000      TX40:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

                                        ; PLINTOCMUX set

000034' 040000  000000          TS40:   TTABLE  (TSTART,0)              ; start up port
000035' 240001  004745'                 TTABLE  (TCHECK,1,[377])        ; check if completed
000036' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet

                                        ; PLINTOCMUX not set

000037' 040000  000100                  TTABLE  (TSTART,100)            ; start up port
000040' 240001  004746'                 TTABLE  (TCHECK,1,[777777777400]) ; check if completed
000041' 340000  000000                  TTABLE  (TEXIT)                 ; exit if error yet
000042' 000000  000000                  TTABLE  (TLAST)                 ; done

                                        ; Error messages

000043' 240000  000045'         MA40:   CALL!TXNOT!MA40LP               ; print subtest description
000044' 000000000000#                   LAST!CALL!TXALL!TTPNT           ; print test data

000045' 200 01 0 00 000020*     MA40LP: MOVE    1,TSTSUB                ; get subtest number
000046' 256 00 0 01 000047'             XCT     MA40L-1(1)              ; print message
000047' 263 17 0 00 000000              RTN                             ; return

000050' 037 00 0 00 004747'     MA40L:  TMSGC   <PLIN not read correctly with PLINTOCMUX asserted>
000051' 037 00 0 00 004762'             TMSGC   <FMTR not read correctly with PLINTOCMUX not asserted>
```

```

                                  ; Microcode:

                                  ;        Read PLIN with PLINTOCMUX asserted

000052' 000000  010000            T40M:  MWORD   <ADDR=0,JMAP,J=1,D=1>                                    ; 0
000053' 001000  000040
000054' 000100  310377                   MWORD   <LDCT,J=31,SDO,OR,D=2,SKCN,MGC=377>                      ; 1
000055' 732000  240300
000056' 000200  020000                   MWORD   <RPCT,J=2,SOA,OR,D=5>                                    ; 2
000057' 435000  000220
000060' 000300  002040                   MWORD   <CONT,SOA,XNOR,B=1,D=3,OENA,SELF,MGC=40>                 ; 3
000061' 473000  403340
000062' 000400  002040                   MWORD   <CONT,SOA,OR,D=1,OENA,SELM,MGC=40>                       ; 4
000063' 431000  002340
000064' 000500  000006                   MWORD   <CONT,SDO,D=1,SELP,MGC=6>                                ; 5
000065' 701000  001340
000066' 000600  000300                   MWORD   <CONT,SDO,OR,B=12,D=2,SELM,MGC=300>                      ; 6
000067' 732005  002340
000070' 000751  000000                   MWORD   <CJP,J=5100,SAB,A=0,B=12,RMIN,D=1,CRY,CENA,CCFZ>;        7
000071' 121405  020460
000072' 001052  000000                   MWORD   <JMAP,J=5200,D=1>                                        ; 10
000073' 001000  000040

                                  ;        Read PLIN without PLINTOCMUX asserted

000074' 010001  010000                   MWORD   <ADDR=100,JMAP,J=101,D=1>                                ; 100
000075' 001000  000040
000076' 010100  310377                   MWORD   <LDCT,J=31,SDO,OR,D=2,SKCN,MGC=377>                      ; 101
000077' 732000  240300
000100' 010201  020000                   MWORD   <RPCT,J=102,SOA,OR,D=5>                                  ; 102
000101' 435000  000220
000102' 010300  002040                   MWORD   <CONT,SOA,XNOR,B=1,D=3,OENA,SELF,MGC=40>                 ; 103
000103' 473000  403340
000104' 010400  002040                   MWORD   <CONT,SOA,OR,D=1,OENA,SELM,MGC=40>                       ; 104
000105' 431000  002340
000106' 010500  000006                   MWORD   <CONT,SDO,D=1,SELP,MGC=6>                                ; 105
000107' 701000  001340
000110' 010600  000200                   MWORD   <CONT,SDO,OR,B=12,D=2,SELM,MGC=200>                      ; 106
000111' 732005  002340
000112' 010751  000000                   MWORD   <CJP,J=5100,SAB,A=1,B=12,RMIN,D=1,CRY,CENA,CCFZ>;        107
000113' 121415  020460
000114' 011052  000000                   MWORD   <JMAP,J=5200,D=1>                                        ; 110
000115' 001000  000040

                                  ; Subroutine - Passed - Write 1 to bit 12 of CSR register

000116' 510000  000000                   MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>                         ; 5100
000117' 742001  000340
000120' 510100  000000                   MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                              ; 5101
000121' 302001  000740
000122' 510200  260000                   MWORD   <LDCT,J=26>                                              ; 5102
000123' 000000  000300
000124' 510351  030000                   MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>                       ; 5103
000125' 102021  000220
```

```
   192  000126' 510400  002004              MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>             ; 5104
   193  000127' 431020  005340
   194  000130' 510551  070040              MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>            ; 5105
   195  000131' 001400  015060
   196  000132' 510651  050000              MWORD   <JMAP,J=5105,D=1>                                 ; 5106
   197  000133' 001000  000040
   198  000134' 510700  000200              MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>                ; 5107
   199  000135' 431010  005340
   200  000136' 511051  102004              MWORD   <JMAP,J=5110,SOA,A=12,OR,D=1,OENA,SELE,MGC=4>     ; 5110
   201  000137' 431120  005040
   202
   203                              ; Subroutine - Failed - Write 1 to bit 13 of CSR register
   204
   205  000140' 520000  000000              MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>                  ; 5200
   206  000141' 742001  000340
   207  000142' 520100  000000              MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                       ; 5201
   208  000143' 302001  000740
   209  000144' 520200  250000              MWORD   <LDCT,J=25>                                       ; 5202
   210  000145' 000000  000300
   211  000146' 520352  030000              MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>                ; 5203
   212  000147' 102021  000220
   213  000150' 520400  002004              MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>             ; 5204
   214  000151' 431020  005340
   215  000152' 520552  070040              MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>            ; 5205
   216  000153' 001400  015060
   217  000154' 520652  050000              MWORD   <JMAP,J=5205,D=1>                                 ; 5206
   218  000155' 001000  000040
   219  000156' 520700  000200              MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>                ; 5207
   220  000157' 431010  005340
   221  000160' 521052  102004              MWORD   <JMAP,J=5210,SOA,A=12,OR,D=1,OENA,SELE,MGC=4>     ; 5210
   222  000161' 431120  005040
   223  000162' 777777  777777              -1
```

```

                                                   ;#*********************************************************************
                                                   ;* Test 41 - CBUF Loaded Correctly
                                                   ;
                                                   ; Description:  This  test  verifies  that  the  CBUF  is  loaded
                                                   ;               correctly at full speed.  The formatter is loaded
                                                   ;               alternately with data from local storage, and the
                                                   ;               2901's.  After each load  the  data  is read back
                                                   ;               and verified.
                                                   ;
                                                   ; Procedure:    KL> Port Clear
                                                   ;               KL> Load test microcode
                                                   ;
                                                   ;               KL> Start microcode at location 0
                                                   ;               UC> Load 2901 R10 with 525252525252
                                                   ;               UC> Load LS location 111 with 252525252525
                                                   ;
                                                   ;               UC> Load formatter with 2901 R10 data, then
                                                   ;                   read it back and verify the data
                                                   ;               UC> Load formatter with LS data, then read
                                                   ;                   it back and verify the data
                                                   ;
                                                   ;               Repeat the above 512 times, on any error set bit
                                                   ;                 13 (ENABLE Complete) in the CSR register, load
                                                   ;                 R11 with type of data (0=R10,1=LS), and load
                                                   ;                 R12 and R13 with correct/actual.
                                                   ;
                                                   ;                 If no error - set bit 12 (DISABLE Complete)
                                                   ;
                                                   ; Failure:      ---
                                                   ;#*********************************************************************

                                                   ; Test data

      000163' 254 00 0 00 000173'                  TSTC41: JRST    TG41                    ; go start test
      000164' 100604  000041                               CBUS!NDMP!NDCB!ZCBUS!41         ; test mask
      000165' 000272' 004775'                              T41M,,[ASCIZ ^CBUF Loaded Correctly^]
      000166' 005002' 005003'                              [EXP MLAST!C24],,[EXP C18,MLAST!C22]
      000167' 000000  000413'                              TSTC42                          ; failure test table
      000170' 000000  000503'                              TSTC43                          ;   ...
      000171' 000000  000572'                              TSTC44
      000172' 777777  777777                               -1

                                                   ; Start test

      000173' 201 00 0 00 000000'                  TG41:   MOVEI   ZA                      ; get address of module start
      000174' 260 17 0 00 000012*                          GO      TRACE                   ; handle trace output
      000175' 201 01 0 00 000272'                          MOVEI   1,T41M                  ; set up microcode address
      000176' 260 17 0 00 000014*                          GO      TLOAD                   ; load/verify it
      000177' 263 17 0 00 000000                           RTN                             ; failed - exit test
```

```


                                        ; Initialization

000200' 400 15 0 00 000000      TL41:   SETZ    ERFLG,                  ; clear error flag
000201' 260 17 0 00 000022*             GO      IPACLR                  ; clear port
000202' 402 00 0 00 000045*             SETZM   TSTSUB                  ; initialize subtest number
000203' 201 06 0 00 000216'             MOVEI   6,TS41                  ; get sstep table address

                                        ; Loop on test execute table entries

000204' 260 17 0 00 000201*     TA41:   GO      IPACLR                  ; clear port
000205' 260 17 0 00 000023*     TB41:   GO      TEXEC                   ; execute table entry
000206' 254 00 0 00 000215'               JRST  TX41                    ; end of sstep table
000207' 254 00 0 00 000205'               JRST  TB41                    ; keep looping after call
000210' 474 15 0 00 000000                SETO  ERFLG,                  ; error occurred

                                        ; Handle error printouts and scope looping

000211' 027 00 0 00 000222'             SCOPER  MA41                    ; print error message
000212' 254 00 0 00 000200'             JRST    TL41                    ; loop on error
000213' 254 00 0 00 000215'             JRST    TX41                    ; altmode exit
000214' 322 15 0 00 000204'             JUMPE   ERFLG,TA41              ; do next sstep table entry

                                        ; End of test

000215' 263 17 0 00 000000      TX41:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

000216' 040000  000000          TS41:   TTABLE  (TSTART,0)              ; start up port
000217' 240040  000000                  TTABLE  (TCHECK,40)             ; check if completed
000220' 340000  000000                  TTABLF  (TEXIT)                 ; exit if error yet
000221' 000000  000000                  TTABLE  (TLAST)                 ; done

                                        ; Error messages

000222' 240000  000224'         MA41:   CALL!TXNOT!MA41LP               ; print subtest description
000223' 000000000000#                   LAST!CALL!TXALL!TTPNT           ; print test data

000224' 400 01 0 00 000000      MA41LP: SETZ    1,                      ; clear AC1
000225' 260 17 0 00 000000*             GO      LDCSR                   ; clear CSR register
000226' 201 01 0 00 005000              MOVEI   1,5000                  ; get address to read registers
000227' 242 01 0 00 000001              LSH     1,1                     ; position properly
000230' 260 17 0 00 000000*             GO      LDRAR                   ; write to RAR
000231' 201 01 0 00 220010              MOVEI   1,MPRUN!SINCYC!TSTEBF   ; set up single step data
000232' 260 17 0 00 000225*             GO      LDCSR                   ; write to CSR register
000233' 260 17 0 00 000000*             GO      RDEBUF                  ; read EBUF data
000234' 202 01 0 00 005725'             MOVEM   1,SAVR11#               ; save R11 data
```

```

000235' 400 01 0 00 000000                SETZ    1,                      ; clear AC1
000236' 260 17 0 00 000232*               GO      LDCSR                   ; clear CSR register
000237' 201 01 0 00 005000                MOVEI   1,5000                  ; get address to read registers
000240' 242 01 0 00 000001                LSH     1,1                     ; position properly
000241' 260 17 0 00 000230*               GO      LDRAR                   ; write to RAR
000242' 201 01 0 00 220010                MOVEI   1,MPRUN!SINCYC!TSTEBF   ; set up single step data
000243' 260 17 0 00 000236*               GO      LDCSR                   ; write to CSR register
000244' 260 17 0 00 000233*               GO      RDEBUF                  ; read EBUF data
000245' 202 01 0 00 005726'               MOVEM   1,SAVR12#               ; save R11 data

000246' 400 01 0 00 000000                SETZ    1,                      ; clear AC1
000247' 260 17 0 00 000243*               GO      LDCSR                   ; clear CSR register
000250' 201 01 0 00 005000                MOVEI   1,5000                  ; get address to read registers
000251' 242 01 0 00 000001                LSH     1,1                     ; position properly
000252' 260 17 0 00 000241*               GO      LDRAR                   ; write to RAR
000253' 201 01 0 00 220010                MOVEI   1,MPRUN!SINCYC!TSTEBF   ; set up single step data
000254' 260 17 0 00 000247*               GO      LDCSR                   ; write to CSR register
000255' 260 17 0 00 000244*               GO      RDEBUF                  ; read EBUF data
000256' 202 01 0 00 005727'               MOVEM   1,SAVR13#               ; save R11 data
000257' 331 01 0 00 005725'               SKIPL   1,SAVR11                ; within range 0-1?
000260' 303 01 0 00 000001                CAILE   1,1
000261' 263 17 0 00 000000                RTN                             ; no - exit
                                          XCT     [TMSGC  <Failing write - 2901 R10 to FMTR>
000262' 256 00 0 01 005023'                        TMSGC  <Failing write - LS111 to FMTR>](1)
000263' 037 00 0 00 005025'               TMSGC   <FMTR Correct:  >
000264' 200 00 0 00 005726'               MOVE    SAVR12
000265' 037 13 0 00 000000                PNTHW
000266' 037 00 0 00 005031'               TMSGC   <       Actual:  >
000267' 200 00 0 00 005727'               MOVE    SAVR13
000270' 037 13 0 00 000000                PNTHW
000271' 263 17 0 00 000000                RTN                             ; return

                                  ; Microcode:

                                  ;       Load 2901 R10 with 525252525252
                                  ;       Load LS location 40 with 252525252525

000272' 000000  010777            T41M:   MWORD   <ADDR=0,JMAP,J=1,SD0,OR,D=2,B=11,SKCN,MGC=777>  ; 0
000273' 732004  640040
000274' 000100  210000                    MWORD   <LDCT,J=21,SOA,AND,D=2,B=10>                    ; 1
000275' 442004  000300
000276' 000200  000000                    MWORD   <CONT,SOB,B=10,PLUS,D=7,CRY>                    ; 2
000277' 307004  000740
000300' 000300  020000                    MWORD   <RPCT,J=2,SOB,B=10,OR,D=7>                      ; 3
000301' 337004  000220
000302' 000410  000000                    MWORD   <LDCT,J=1000,SOB,PLUS,B=10,CRY,D=7>             ; 4
000303' 307004  000700

000304' 000500  002040                    MWORD   <CONT,SOA,XNOR,A=10,D=1,OENA,LDLM,MGC=40>       ; 5
000305' 471100  230340

                                  ;       Load formatter with 2901 R10 data, then read and verify

000306' 000600  002040                    MWORD   <CONT,SOA,OR,A=10,B=12,D=3,OENA,SELF,MGC=40>    ; 6
```

```
000307' 433105  003340
000310' 000700  000200                  MWORD   <CONT,SDO,OR,B=13,D=2,SELM,MGC=200>                     ; 7
000311' 732005  402340
000312' 001000  120000                  MWORD   <CJP,J=12,SAB,A=12,B=13,SMIN,CRY,CENA,CCFZ>             ; 10
000313' 110525  420460
000314' 001101  000000                  MWORD   <JMAP,J=100,D=1>                                        ; 11
000315' 001000  000040

                                ;       Load formatter with LS data, then read read and verify

000316' 001200  000040                  MWORD   <CONT,SDO,OR,B=12,D=2,RDLM,SELF,MGC=40>                 ; 12
000317' 732005  223340
000320' 001300  000200                  MWORD   <CONT,SDO,OR,B=13,D=2,SELM,MGC=200>                     ; 13
000321' 732005  402340
000322' 001400  160000                  MWORD   <CJP,J=16,SAB,A=12,B=13,SMIN,CRY,CENA,CCFZ>             ; 14
000323' 110525  420460
000324' 001501  010000                  MWORD   <JMAP,J=101,D=1>                                        ; 15
000325' 001000  000040

                                ;       Now loop till done

000326' 001600  050000                  MWORD   <RPCT,J=5,D=1>                                          ; 16
000327' 001000  000220
000330' 001751  000000                  MWORD   <JMAP,J=5100,D=1>                                       ; 17
000331' 001000  000040

                                ;       Failed loading with R10 data (100)
                                ;       Failed loading with LS data  (101)

000332' 010052  000000                  MWORD   <ADDR=100,JMAP,J=5200,SOA,AND,B=11,D=2>                 ; 100
000333' 442004  400040
000334' 010100  000000                  MWORD   <CONT,SOA,AND,B=11,D=2>                                 ; 101
000335' 442004  400340
000336' 010252  000000                  MWORD   <JMAP,J=5200,SOB,PLUS,D=2,B=11,CRY>                     ; 102
000337' 302004  400440

                                ; Routine to read 2901 Register 11.12.13

000340' 500050  012004                  MWORD   <ADDR=5000,JMAP,J=5001,SOA,A=11,OR,D=1,OENA,SELE,MGC=4>
000341' 431110  005040
000342' 500150  022004                  MWORD   <JMAP,J=5002,SOA,A=12,OR,D=1,OENA,SELE,MGC=4>
000343' 431120  005040
000344' 500250  022004                  MWORD   <JMAP,J=5002,SOA,A=13,OR,D=1,OENA,SELE,MGC=4>
000345' 431130  005040

                                ; Subroutine - Passed - Write 1 to bit 12 of CSR register

000346' 510000  000000                  MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>        ; 5100
000347' 742001  000340
000350' 510100  000000                  MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5101
000351' 302001  000740
000352' 510200  260000                  MWORD   <LDCT,J=26>                             ; 5102
000353' 000000  000300
000354' 510351  030000                  MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>      ; 5103
000355' 102021  000220
```

```
000356' 510400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5104
000357' 431020  005340
000360' 510551  070040          MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>  ; 5105
000361' 001400  015060
000362' 510651  050000          MWORD   <JMAP,J=5105,D=1>                       ; 5106
000363' 001000  000040
000364' 510700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5107
000365' 431010  005340
000366' 511051  100000          MWORD   <JMAP,J=5110,D=1>                       ; 5110
000367' 001000  000040

                        ; Subroutine - Failed - Write 1 to bit 13 of CSR register

000370' 520000  000000          MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>        ; 5200
000371' 742001  000340
000372' 520100  000000          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5201
000373' 302001  000740
000374' 520200  250000          MWORD   <LDCT,J=25>                             ; 5202
000375' 000000  000300
000376' 520352  030000          MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>      ; 5203
000377' 102021  000220
000400' 520400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5204
000401' 431020  005340
000402' 520552  070040          MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>  ; 5205
000403' 001400  015060
000404' 520652  050000          MWORD   <JMAP,J=5205,D=1>                       ; 5206
000405' 001000  000040
000406' 520700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5207
000407' 431010  005340
000410' 521052  100000          MWORD   <JMAP,J=5210,D=1>                       ; 5210
000411' 001000  000040
000412' 777777  777777          -1
```

```
                                          ;#*********************************************************************
                                          ;* Test 42 - PMUX Selected by MPCBUFTOPLOUT
                                          ;
                                          ; Description:  This test verifies the select line  of  the PMUX.
                                          ;               A data pattern is written to  the  formatter  and
                                          ;               the complement of that data pattern is written to
                                          ;               the PLOUT with MPCBUFTOPLOUT signal asserted. The
                                          ;               data written to the PLOUT is then verified.
                                          ;
                                          ; Procedure:    KL> Port Clear
                                          ;               KL> Load test microcode
                                          ;
                                          ;               KL> Single step 5 times from location 0
                                          ;               UC> Load formatter with 0's
                                          ;               UC> Load PLOUT with 1's
                                          ;               UC> Read PLOUT and put into EBUF
                                          ;               KL> Verify EBUF data
                                          ;
                                          ;               KL> Single step 6 times from location 10
                                          ;               UC> Load formatter with 1's
                                          ;               UC> Load PLOUT with 0's
                                          ;               UC> Read PLOUT and put into EBUF
                                          ;               KL> Verify EBUF data
                                          ;
                                          ; Failure:      ---
                                          ;#*********************************************************************

                                          ; Test data

 000413' 254 00 0 00 000423'              TSTC42: JRST    TG42                            ; go start test
 000414' 100604  000042                           CBUS!NDMP!NDCB!ZCBUS!42                 ; test mask
 000415' 000454' 005035'                          T42M,,[ASCIZ ^PMUX Selected by MPCBUFTOPLOUT^]
 000416' 005044' 000000                           [EXP C8,MLAST!C22],,0
 000417' 000000  000503'                          TSTC43                                  ; failure test table
 000420' 000000  000572'                          TSTC44                                  ;   ...
 000421' 000000  002141'                          TSTC52
 000422' 777777  777777                           -1

                                          ; Start test

 000423' 201 00 0 00 000000'              TG42:   MOVEI   ZA                              ; get address of module start
 000424' 260 17 0 00 000174*                      GO      TRACE                           ; handle trace output
 000425' 201 01 0 00 000454'                      MOVEI   1,T42M                          ; set up microcode address
 000426' 260 17 0 00 000176*                      GO      TLOAD                           ; load/verify it
 000427' 263 17 0 00 000000                       RTN                                     ; failed - exit test

                                          ; Initialization

 000430' 400 15 0 00 000000               TL42:   SETZ    ERFLG,                          ; clear error flag
 000431' 260 17 0 00 000204*                      GO      IPACLR                          ; clear port
 000432' 402 00 0 00 000202*                      SETZM   TSTSUB                          ; initialize subtest number
 000433' 201 06 0 00 000445'                      MOVEI   6,TS42                          ; get sstep table address
```

```

                                  ; Loop on single step table entries

000434' 260 17 0 00 000000*       TA42:   GO      BEXEC                   ; execute table entry
000435' 254 00 0 00 000444'               JRST    TX42                    ; end of sstep table
000436' 254 00 0 00 000434'               JRST    TA42                    ; keep looping after call
000437' 474 15 0 00 000000                SETO    ERFLG,                  ; error - EBUF data incorrect

                                  ; Handle error printouts and scope looping

000440' 027 00 0 00 000452'               SCOPER  MA42                    ; print error message
000441' 254 00 0 00 000430'               JRST    TL42                    ; loop on error
000442' 254 00 0 00 000444'               JRST    TX42                    ; altmode exit
000443' 322 15 0 00 000434'               JUMPE   ERFLG,TA42              ; do next sstep table entry

                                  ; End of test

000444' 263 17 0 00 000000        TX42:   RTN                             ; return

                                  ; Sstep table, with EBUF data, as:  (CMD,#Steps,Start addr,End Addr,EBUF data)

000445' 100500  000004            TS42:   ATABLE  (SSSTRT,5,0,4,377)
000446' 000000  000377
000447' 100600  100015                    ATABLE  (SSSTRT,6,10,15,0)
000450' 000000  000000
000451' 000000  000000                    ATABLE  (SSLAST)

                                  ; Error messages

000452' 140000  005046'           MA42:   MSG!TXNOT![ASCIZ /Incorrect data loaded into PLOUT (read and put into EBUF)/]
000453' 000000000000#                     LAST!CALL!TXALL!BBPNT           ; print sequencer data

                                  ; Microcode:

000454' 000000  012040            T42M:   MWORD   <ADDR=0,JMAP,J=1,SOA,AND,D=3,OENA,SELF,MGC=40>  ; 0
000455' 443000  003040
000456' 000100  002040                    MWORD   <CONT,SOA,XNOR,D=1,OENA,SELM,MGC=40>            ; 1
000457' 471000  002340
000460' 000200  000006                    MWORD   <CONT,SDO,D=1,SELP,MGC=6>                       ; 2
000461' 701000  001340
000462' 000300  000300                    MWORD   <CONT,SDO,OR,B=2,D=2,SELM,MGC=300>              ; 3
000463' 732001  002340
000464' 000400  042004                    MWORD   <JMAP,J=4,SOA,A=2,D=1,OENA,SELE,MGC=4>          ; 4
000465' 401020  005040

000466' 001000  110000                    MWORD   <ADDR=10,JMAP,J=11,SOA,AND,D=2>                 ; 10
000467' 442000  000040
000470' 001100  002040                    MWORD   <CONT,SOA,XNOR,D=3,OENA,SELF,MGC=40>            ; 11
000471' 473000  003340
000472' 001200  002040                    MWORD   <CONT,SOA,XNOR,D=1,OENA,SELM,MGC=40>            ; 12
000473' 471000  002340
000474' 001300  000006                    MWORD   <CONT,SDO,D=1,SELP,MGC=6>                       ; 13
000475' 701000  001340
000476' 001400  000300                    MWORD   <CONT,SDO,OR,B=2,D=2,SELM,MGC=300>              ; 14
000477' 732001  002340
```

```
   572  000500' 001500  152004                      MWORD   <JMAP,J=15,SOA,A=2,D=1,OENA,SELE,MGC=4>         ; 15
   573  000501' 401020  005040
   574  000502' 777777  777777                      -1
```

```

                                       ;#*********************************************************************
                                       ;* Test 43 - PMUX Selected by MPCBUFTOPLOUT/MPZEROLFTNIB
                                       ;
                                       ; Description:  This test verifies the select line  of  the PMUX.
                                       ;               A data pattern is written to  the  formatter  and
                                       ;               the complement of that data pattern is written to
                                       ;               the  PLOUT  with  MPCBUFTOPLOUT  and MPZEROLFTNIB
                                       ;               signals asserted.  The  data written to the PLOUT
                                       ;               is then verified.
                                       ;
                                       ; Procedure:    KL> Port Clear
                                       ;               KL> Load test microcode
                                       ;
                                       ;               KL> Single step 5 times from location 0
                                       ;               UC> Load formatter with 0's
                                       ;               UC> Load PLOUT with 1's
                                       ;               UC> Read PLOUT and put into EBUF
                                       ;               KL> Verify EBUF data
                                       ;
                                       ;               KL> Single step 6 times from location 10
                                       ;               UC> Load formatter with 1's
                                       ;               UC> Load PLOUT with 0's
                                       ;               UC> Read PLOUT and put into EBUF
                                       ;               KL> Verify EBUF data
                                       ;
                                       ; Failure:      ---
                                       ;#*********************************************************************

                                       ; Test data

000503' 254 00 0 00 000512'            TSTC43: JRST    TG43                    ; go start test
000504' 100604  000043                         CBUS!NDMP!NDCB!ZCBUS!43         ; test mask
000505' 000543' 005062'                        T43M,,[ASCIZ ^PMUX Selected by MPCBUFTOPLOUT/MPZEROLFTNIB^]
000506' 005044' 000000                         [EXP C8,MLAST!C22],,0
000507' 000000  000572'                        TSTC44                          ; failure test table
000510' 000000  002141'                        TSTC52                          ;   ...
000511' 777777  777777                         -1

                                       ; Start test

000512' 201 00 0 00 000000'            TG43:   MOVEI   ZA                      ; get address of module start
000513' 260 17 0 00 000424*                    GO      TRACE                   ; handle trace output
000514' 201 01 0 00 000543'                    MOVEI   1,T43M                  ; set up microcode address
000515' 260 17 0 00 000426*                    GO      TLOAD                   ; load/verify it
000516' 263 17 0 00 000000                     RTN                             ; failed - exit test

                                       ; Initialization

000517' 400 15 0 00 000000             TL43:   SETZ    ERFLG,                  ; clear error flag
000520' 260 17 0 00 000431*                    GO      IPACLR                  ; clear port
000521' 402 00 0 00 000432*                    SETZM   TSTSUB                  ; initialize subtest number
000522' 201 06 0 00 000534'                    MOVEI   6,TS43                  ; get sstep table address
```

```

                                   ; Loop on single step table entries

000523' 260 17 0 00 000434*        TA43:   GO      BEXEC                   ; execute table entry
000524' 254 00 0 00 000533'                JRST    TX43                    ; end of sstep table
000525' 254 00 0 00 000523'                JRST    TA43                    ; keep looping after call
000526' 474 15 0 00 000000                 SETO    ERFLG,                  ; error - EBUF data incorrect

                                   ; Handle error printouts and scope looping

000527' 027 00 0 00 000541'                SCOPER  MA43                    ; print error message
000530' 254 00 0 00 000517'                JRST    TL43                    ; loop on error
000531' 254 00 0 00 000533'                JRST    TX43                    ; altmode exit
000532' 322 15 0 00 000523'                JUMPE   ERFLG,TA43              ; do next sstep table entry

                                   ; End of test

000533' 263 17 0 00 000000         TX43:   RTN                             ; return

                                   ; Sstep table, with EBUF data, as:  (CMD,#Steps,Start addr,End Addr,EBUF data)

000534' 100500  000004             TS43:   ATABLE  (SSSTRT,5,0,4,377)
000535' 000000  000377
000536' 100600  100015                     ATABLE  (SSSTRT,6,10,15,0)
000537' 000000  000000
000540' 000000  000000                     ATABLE  (SSLAST)

                                   ; Error messages

000541' 140000  005046'            MA43:   MSG!TXNOT![ASCIZ /Incorrect data loaded into PLOUT (read and put into EBUF)/]
000542' 000000000000#                      LAST!CALL!TXALL!BBPNT           ; print sequencer data

                                   ; Microcode:

000543' 000000  012041             T43M:   MWORD   <ADDR=0,JMAP,J=1,SOA,AND,D=3,OENA,SELF,MGC=41>  ; 0
000544' 443000  003040
000545' 000100  002040                     MWORD   <CONT,SOA,XNOR,D=1,OENA,SELM,MGC=40>            ; 1
000546' 471000  002340
000547' 000200  000006                     MWORD   <CONT,SDO,D=1,SELP,MGC=6>                       ; 2
000550' 701000  001340
000551' 000300  000300                     MWORD   <CONT,SDO,OR,B=2,D=2,SELM,MGC=300>              ; 3
000552' 732001  002340
000553' 000400  042004                     MWORD   <JMAP,J=4,SOA,A=2,D=1,OENA,SELE,MGC=4>          ; 4
000554' 401020  005040

000555' 001000  110000                     MWORD   <ADDR=10,JMAP,J=11,SOA,AND,D=2>                 ; 10
000556' 442000  000040
000557' 001100  002041                     MWORD   <CONT,SOA,XNOR,D=3,OENA,SELF,MGC=41>            ; 11
000560' 473000  003340
000561' 001200  002040                     MWORD   <CONT,SOA,XNOR,D=1,OENA,SELM,MGC=40>            ; 12
000562' 471000  002340
000563' 001300  000006                     MWORD   <CONT,SDO,D=1,SELP,MGC=6>                       ; 13
000564' 701000  001340
000565' 001400  000300                     MWORD   <CONT,SDO,OR,B=2,D=2,SELM,MGC=300>              ; 14
000566' 732001  002340
```

```
   683  000567' 001500  152004                     MWORD   <JMAP,J=15,SOA,A=2,D=1,OENA,SELE,MGC=4>         ; 15
   684  000570' 401020  005040
   685  000571' 777777  777777                     -1
```

```
                                   ;#**********************************************************************
                                   ;* Test 44 - DMUX Selected Correctly
                                   ;
                                   ; Description:  Verify that the DMUX can be selected correctly by
                                   ;               either  asserting or not asserting 'MPCBUFTOFMTR'
                                   ;               signal.
                                   ;
                                   ; Procedure:    KL> Port Clear
                                   ;               KL> Set up a CCW list to transfer 2 words
                                   ;               KL> Start up port
                                   ;
                                   ;               UC> Read word from CBUS into CBIN
                                   ;               UC> Put 0's on the MBUS and load the formatter
                                   ;               UC> Verify data 0's
                                   ;               UC> Put 0's on the MBUS and load the formatter
                                   ;                   with 'MPCBUFTOFMTR' set
                                   ;               UC> Verify data 1's
                                   ;
                                   ;               UC> Read word from CBUS into CBIN
                                   ;               UC> Put 1's on the MBUS and load the formatter
                                   ;               UC> Verify data 1's
                                   ;               UC> Put 1's on the MBUS and load the formatter
                                   ;                   with 'MPCBUFTOFMTR' set
                                   ;               UC> Verify data 0's
                                   ;
                                   ;               UC> if passed - set bit 12 (DISABLE Complete)
                                   ;               UC> if failed - set bit 13 (ENABLE Complete)
                                   ;               Read CSR and determine test disposition
                                   ;
                                   ; Failure:      ---
                                   ;#**********************************************************************

                                   ; Test data

000572' 254 00 0 00 000603'        TSTC44: JRST    TG44                    ; go start test
000573' 100604  000044                     CBUS!NDMP!NDCB!ZCBUS!44         ; test mask
000574' 000654' 005073'                    T44M,,[ASCIZ ^DMUX Selected Correctly^]
000575' 005100' 005102'                    [EXP C8,MLAST!C18],,[EXP MLAST!C20]
000576' 000000  001005'                    TSTC45                          ; failure test table
000577' 000000  001172'                    TSTC46                          ;   ...
000600' 000000  001326'                    TSTC47
000601' 000000  002141'                    TSTC52
000602' 777777  777777                     -1

                                   ; Start test

000603' 201 00 0 00 000000'        TG44:   MOVEI   ZA                      ; get address of module start
000604' 260 17 0 00 000513*                GO      TRACE                   ; handle trace output
000605' 201 01 0 00 000654'                MOVEI   1,T44M                  ; set up microcode address
000606' 260 17 0 00 000515*                GO      TLOAD                   ; load/verify it
000607' 263 17 0 00 000000                 RTN                             ; failed - exit test
```

```

                                 ; Initialization

000610' 400 15 0 00 000000       TL44:    SETZ     ERFLG,                       ; clear error flag
000611' 201 06 0 00 000626'               MOVEI    6,TS44                       ; get test table address
000612' 402 00 0 00 000521*               SETZM    TSTSUB                       ; initialize subtest number
000613' 402 00 0 00 000000*               SETZM    CSTATF                       ; clear check logout status flag

                                 ; Loop on test execute table entries

000614' 260 17 0 00 000520*      TA44:    GO       IPACLR                       ; clear port
000615' 260 17 0 00 000205*      TB44:    GO       TEXEC                        ; execute table entry
000616' 254 00 0 00 000625'                 JRST   TX44                         ; end of test execute table
000617' 254 00 0 00 000615'                 JRST   TB44                         ; keep looping after call
000620' 474 15 0 00 000000                  SETO   ERFLG,                       ; error occurred

                                 ; Handle error printouts and scope looping

000621' 027 00 0 00 000652'               SCOPER   MA44                         ; print error message
000622' 254 00 0 00 000610'               JRST     TL44                         ; loop on error
000623' 254 00 0 00 000625'               JRST     TX44                         ; altmode exit
000624' 322 15 0 00 000614'               JUMPE    ERFLG,TA44                   ; do next test execute table entry

                                 ; End of test

000625' 263 17 0 00 000000       TX44:    RTN                                   ; return

                                 ; Test Execute Table, as:  (CMD,parameters)

000626' 140000  000633'          TS44:    TTABLE   (TCALL,TS44DA)               ; set up data transfer
000627' 040000  005002                    TTABLE   (TSTART,5002)                ; start up port
000630' 240002  000000                    TTABLE   (TCHECK,2)                   ; check if completed
000631' 340000  000000                    TTABLE   (TEXIT)                      ; exit if error yet
000632' 000000  000000                    TTABLE   (TLAST)                      ; end of table

                                 ; Set up data transfer

000633' 201 01 0 00 005000       TS44DA: MOVEI     1,5000                       ; get start address
000634' 242 01 0 00 000001               LSH       1,1                          ; position correctly
000635' 260 17 0 00 000252*              GO        LDRAR                        ; load the register
000636' 201 01 0 00 020010               MOVEI     1,SINCYC!MPRUN               ; get single step data
000637' 260 17 0 00 000254*              GO        LDCSR                        ; single step once
000640' 260 17 0 00 000637*              GO        LDCSR                        ; single step once more
000641' 476 00 0 00 000000*              SETOM     BUFF                         ; initialize data words
000642' 402 00 0 00 000000#              SETZM     BUFF+1                       ;   ...
000643' 200 01 0 00 005103'              MOVE      1,[CBUFSZ,,CBUF]             ; get pointer to control buffer
000644' 260 17 0 00 000000*              GO        CHINIT                       ; initialize software
000645' 551 01 0 00 000641*              HRRZI     1,BUFF                       ; buffer address
000646' 201 02 0 00 000002               MOVEI     2,2                          ; word count
000647' 400 03 0 00 000000               SETZ      3,                           ; clear zero fill specifier
000650' 260 17 0 00 000000*              GO        GENCCW                       ; generate a CCW list
000651' 263 17 0 00 000000               RTN                                    ; return
```

```

                                        ; Error messages

000652' 140000  005104'                 MA44:   MSG!TXNOT![ASCIZ /Formatter not loaded correctly through DMUX/]
000653' 000000000000#                           LAST!CALL!TXALL!TTPNT           ; print test data

                                        ; Microcode

                                        ;       Load fmtr through CBUF 0's, read fmtr and verify

000654' 000001  000000                  T44M:   MWORD   <ADDR=0,CJS,J=100,SOA,AND,B=3,D=2>      ; 0
000655' 442001  400020
000656' 000100  002040                          MWORD   <CONT,SOA,OR,A=3,D=1,OENA,SELF,MGC=40>  ; 1
000657' 431030  003340
000660' 000200  000200                          MWORD   <CONT,SDO,OR,B=2,D=2,SELM,MGC=200>      ; 2
000661' 732001  002340
000662' 000300  050000                          MWORD   <CJP,J=5,SOA,A=2,OR,D=1,CENA,CCFZ>      ; 3
000663' 431420  020060
000664' 000452  000000                          MWORD   <JMAP,J=5200,D=1>                       ; 4
000665' 001000  000040

                                        ;       Put 0's on MBUS, load fmtr with CBIN, read fmtr and verify

000666' 000500  002004                          MWORD   <CONT,SOA,OR,A=3,D=1,OENA,SELC,MGC=4>   ; 5
000667' 431030  004340
000670' 000600  000200                          MWORD   <CONT,SDO,OR,B=2,D=2,SELM,MGC=200>      ; 6
000671' 732001  002340
000672' 000700  110000                          MWORD   <CJP,J=11,SOA,A=2,XNOR,D=1,CENA,CCFZ>   ; 7
000673' 471420  020060
000674' 001052  000000                          MWORD   <JMAP,J=5200,D=1>                       ; 10
000675' 001000  000040

                                        ;       Load fmtr through CBUF 1's, read fmtr and verify

000676' 001101  010000                          MWORD   <CJS,J=101,SOB,XNOR,B=3,D=2>            ; 11
000677' 372001  400020
000700' 001200  002040                          MWORD   <CONT,SOA,OR,A=3,D=1,OENA,SELF,MGC=40>  ; 12
000701' 431030  003340
000702' 001300  000200                          MWORD   <CONT,SDO,OR,B=2,D=2,SELM,MGC=200>      ; 13
000703' 732001  002340
000704' 001400  160000                          MWORD   <CJP,J=16,SOA,A=2,XNOR,D=1,CENA,CCFZ>   ; 14
000705' 471420  020060
000706' 001552  000000                          MWORD   <JMAP,J=5200,D=1>                       ; 15
000707' 001000  000040

                                        ;       Put 1's on MBUS, load fmtr with CBIN, read fmtr and verify

000710' 001600  002004                          MWORD   <CONT,SOA,XNOR,A=3,D=1,OENA,SELC,MGC=4> ; 16
000711' 471030  004340
000712' 001700  000200                          MWORD   <CONT,SDO,OR,B=2,D=2,SELM,MGC=200>      ; 17
000713' 732001  002340
000714' 002000  220000                          MWORD   <CJP,J=22,SOA,A=2,OR,D=1,CENA,CCFZ>     ; 20
000715' 431420  020060
000716' 002152  000000                          MWORD   <JMAP,J=5200,D=1>                       ; 21
000717' 001000  000040
```

```


                                  ;       All done

000720' 002251  000000                    MWORD   <JMAP,J=5100,D=1>                       ; 22
000721' 001000  000040

                                  ;       Read a word from CBUS into CBIN

000722' 010000  000200                    MWORD   <ADDR=100,CONT,D=1,SELC,MGC=200>        ; 100
000723' 001000  004340
000724' 010101  030000                    MWORD   <CJP,J=103,D=1,CENA,CCCA>               ; 101
000725' 001400  000060
000726' 010201  010000                    MWORD   <JMAP,J=101,D=1>                        ; 102
000727' 001000  000040
000730' 010300  000000                    MWORD   <CRTN,D=1>                              ; 103
000731' 001000  000240

                                  ;       Initialization steps

000732' 500050  010100                    MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>    ; 5000
000733' 000000  004040
000734' 500150  010020                    MWORD   <JMAP,J=5001,SELM,MGC=20>               ; 5001
000735' 000[illegible]  002040

000736' 50[illegible]  000000                    MWORD   <JZ>                                    ; 5002
000737' 000000  000000

                                  ; Subroutine - Passed - Write 1 to bit 12 of CSR register

000740' 510000  000000                    MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>        ; 5100
000741' 742001  000340
000742' 510100  000000                    MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5101
000743' 302001  000740
000744' 510200  260000                    MWORD   <LDCT,J=26>                             ; 5102
000745' 000000  000300
000746' 510351  030000                    MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>      ; 5103
000747' 102021  000220
000750' 510400  002004                    MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5104
000751' 431020  005340
000752' 510551  070040                    MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>  ; 5105
000753' 001400  015060
000754' 510651  050000                    MWORD   <JMAP,J=5105,D=1>                       ; 5106
000755' 001000  000040
000756' 510700  000200                    MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5107
000757' 431010  005340
000760' 511051  100000                    MWORD   <JMAP,J=5110,D=1>                       ; 5110
000761' 001000  000040

                                  ; Subroutine - Failed - Write 1 to bit 13 of CSR register

000762' 520000  000000                    MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>        ; 5200
000763' 742001  000340
000764' 520100  000000                    MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5201
000765' 302001  000740
000766' 520200  250000                    MWORD   <LDCT,J=25>                             ; 5202
```

```
000767' 000000  000300
000770' 520352  030000          MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>          ; 5203
000771' 102021  000220
000772' 520400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>       ; 5204
000773' 431020  005340
000774' 520552  070040          MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>      ; 5205
000775' 001400  015060
000776' 520652  050000          MWORD   <JMAP,J=5205,D=1>                           ; 5206
000777' 001000  000040
001000' 520700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>          ; 5207
001001' 431010  005340
001002' 521052  100000          MWORD   <JMAP,J=5210,D=1>                           ; 5210
001003' 001000  000040
001004' 777777  777777          -1
```

```
                                         ;#*********************************************************************
                                         ;* Test 45 - CBUS Parity Checking - CBUS to CBUS Module
                                         ;
                                         ; Description:  This test reads a pattern of  floating  1's  over
                                         ;               the CBUS from KL memory, verifying with each word
                                         ;               transfered that a parity error has not occurred.
                                         ;
                                         ; Procedure:    KL> Port Clear
                                         ;               KL> Set up a CCW list to transfer 36 words
                                         ;               KL> Start up port
                                         ;               UC> Initialize word count
                                         ;               UC> Read a word from CBUS and put in 2901 R3
                                         ;               UC> Repeat until 'Last Wd' is seen:
                                         ;                      Read a word
                                         ;                      Verify that a parity error has not occurred
                                         ;                        If error - put word in error into R3
                                         ;                                 - put word count in EBUF
                                         ;                                 - abort test
                                         ;                        If ok - do next word
                                         ;               UC> Assert STOP and STORE and put word count into EBUF
                                         ;               KL> Stop microcode, read EBUF - verify word count
                                         ;               KL> If stopped at wrong location, read R3 to get
                                         ;                   actual data that resulted in parity error.
                                         ;               KL> Verify data transfer completed ok.
                                         ;
                                         ; Failure:      ---
                                         ;#*********************************************************************

                                         ; Test data

 001005' 254 00 0 00 001014'             TSTC45: JRST    TG45                        ; go start test
 001006' 100604  000045                          CBUS!NDMP!NDCB!ZCBUS!45             ; test mask
 001007' 001111' 005115'                         T45M,,[ASCIZ ^CBUS Parity Checking - CBUS to CBUS Module^]
 001010' 005126' 005130'                         [EXP C17,MLAST!C9],,[EXP MLAST!C7]
 001011' 000000  001172'                         TSTC46                              ; failure test table
 001012' 000000  001326'                         TSTC47                              ;   ...
 001013' 777777  777777                          -1

                                         ; Start test

 001014' 201 00 0 00 000000'             TG45:   MOVEI   ZA                          ; get address of module start
 001015' 260 17 0 00 000604*                     GO      TRACE                       ; handle trace output
 001016' 201 01 0 00 001111'                     MOVEI   1,T45M                      ; set up microcode address
 001017' 260 17 0 00 000606*                     GO      TLOAD                       ; load/verify it
 001020' 263 17 0 00 000000                      RTN                                 ; failed - exit test

                                         ; Initialization

 001021' 400 15 0 00 000000              TL45:   SETZ    ERFLG,                      ; clear error flag
 001022' 201 06 0 00 001036'                     MOVEI   6,TS45                      ; get test table address
 001023' 402 00 0 00 000612*                     SETZM   TSTSUB                      ; initialize subtest number
```

```
                                    ; Loop on test execute table entries

001024' 260 17 0 00 000614*         TA45:   GO      IPACLR                  ; clear port
001025' 260 17 0 00 000000*         TB45:   GO      CEXEC                   ; execute table entry
001026' 254 00 0 00 001035'                 JRST    TX45                    ; end of test execute table
001027' 254 00 0 00 001025'                 JRST    TB45                    ; keep looping after call
001030' 474 15 0 00 000000                  SETO    ERFLG,                  ; error occurred

                                    ; Handle error printouts and scope looping

001031' 027 00 0 00 001052'                 SCOPER  MA45                    ; print error message
001032' 254 00 0 00 001021'                 JRST    TL45                    ; loop on error
001033' 254 00 0 00 001035'                 JRST    TX45                    ; altmode exit
001034' 322 15 0 00 001024'                 JUMPE   ERFLG,TA45              ; do next test execute table entry

                                    ; End of test

001035' 263 17 0 00 000000          TX45:   RTN                             ; return

                                    ; Data Test Execute Table, as:  (CMD,parameters)

001036' 100120  000044              TS45:   CTABLE  (CSETWR,120,^D36)       ; set up write (36 words)
001037' 200000  001000                      CTABLE  (CSTART,1000)           ; start up port
001040' 340010  000000                      CTABLE  (CWAIT,10,0)            ; check if completed
001041' 240000  001043'                     CTABLE  (CCALL,TS45CH)          ; check results
001042' 000000  000000                      CTABLE  (CLAST)                 ; end of table

                                    ; Check results

001043' 260 17 0 00 001055'         TS45CH: GO      MA45CH                  ; get additional data
001044' 200 01 0 00 000000*                 MOVE    1,CEBUFA                ; get word count
001045' 302 01 0 00 000044                  CAIE    1,^D36                  ; correct?
001046' 474 15 0 00 000000                  SETO    ERFLG,                  ; no - set error flag
001047' 350 00 0 17 777772                  AOS     -6(P)                   ; set up error return
001050' 350 00 0 17 777772                  AOS     -6(P)                   ; set up error return
001051' 263 17 0 00 000000                  RTN                             ; return

                                    ; Error messages

001052' 140000  005131'             MA45:   MSG!TXNOT![ASCIZ /Parity error detected in data transfer/]
001053' 240000  001055'                     CALL!TXNOT!MA45CH               ; check other results
001054' 000000000000#                       LAST!CALL!TXALL!CCPNT           ; print test data
```

```
                                        ; Check results - first the word count left in the EBUF

001055' 322 15 0 00 001067'     MA45CH: JUMPE   ERFLG,MA45CO                   ; no error yet? - continue
001056' 200 01 0 00 001044*             MOVE    1,CEBUFA                       ; get word count
001057' 306 01 0 00 000044              CAIN    1,^D36                         ; correct?
001060' 254 00 0 00 001055'             JRST    MA45CH0                        ; yes - continue
001061' 331 00 0 00 001056*             SKIPL   CEBUFA
                                        JRST    [TMSGC  <Port word count error:  >
                                                 TMSGC  <  Correct:    36.>
                                                 TMSGC  <  Actual:>
                                                 PDEC   6,CEBUFA                ; print it
                                                 TMSG   <.>
001062' 254 00 0 00 005157'                      JRST   MA45CO]
001063' 037 00 0 00 005165'             TMSGC   <Parity error occurred on word #>
001064' 210 00 0 00 001061*             MOVN    CEBUFA
001065' 037 15 0 00 000000              PNTDEC
001066' 037 00 0 00 005156'             TMSG    <.>

                                        ; Then the correct/actual data being read

001067' 400 01 0 00 000000      MA45CO: SETZ    1,                             ; ensure the port is
001070' 260 17 0 00 000640*             GO      LDCSR                          ;   stopped
001071' 201 01 0 00 000200              MOVEI   1,200                          ; get address to get R3
001072' 242 01 0 00 000001              LSH     1,1                            ; position correctly
001073' 260 17 0 00 000635*             GO      LDRAR                          ; load RAR register
001074' 201 01 0 00 000010              MOVEI   1,MPRUN                        ; set up initial CSR data
001075' 260 17 0 00 001070*             GO      LDCSR                          ; write CSR register
001076' 201 01 0 00 200000              MOVEI   1,TSTEBF                       ; set up CSR to read EBUF
001077' 260 17 0 00 001075*             GO      LDCSR                          ; write it
001100' 260 17 0 00 000255*             GO      RDEBUF                         ; get EBUF data (actual)
001101' 202 01 0 00 005722'             MOVEM   1,SAVDA1#                      ; save it
001102' 400 01 0 00 000000              SETZ    1,                             ; ensure the port is
001103' 260 17 0 00 001077*             GO      LDCSR                          ;   stopped
001104' 322 15 0 00 005174'             JUMPE   ERFLG,[RTN]                    ; no error yet? - continue
001105' 037 00 0 00 005175'             TMSGC   <Word with parity error:  >
001106' 200 00 0 00 005722'             MOVE    SAVDA1                         ; get data
001107' 037 13 0 00 000000              PNTHW                                  ; print it
001110' 263 17 0 00 000000              RTN                                    ; return
```

```

                                ; Microcode - Read each word

001111' 000000  010000          T45M:   MWORD   <ADDR=0,JMAP,J=1,SOB,AND,B=5,D=2>               ; 0
001112' 342002  400040
001113' 000101  000000                  MWORD   <CJS,J=100,D=1>                                 ; 1
001114' 001000  000020
001115' 000200  040000                  MWORD   <JMAP,J=4,D=1>                                  ; 2
001116' 001000  000040
001117' 000301  010000                  MWORD   <CJS,J=101,D=1>                                 ; 3
001120' 001000  000020
001121' 000400  070000                  MWORD   <CJP,J=7,D=1,CENA,CCCP>                         ; 4
001122' 001400  140060
001123' 000500  120000                  MWORD   <CJP,J=12,D=1,CENA,CCLW>                        ; 5
001124' 001400  170060
001125' 000600  030000                  MWORD   <JMAP,J=3>                                      ; 6
001126' 000000  000040

                                ;       Parity error occurred - set STOP and STORE and write word count to EBUF

001127' 000700  000140                  MWORD   <CONT,SELC,MGC=140>                             ; 7
001130' 000000  004340
001131' 001000  002004                  MWORD   <CONT,SOA,XNOR,A=5,D=1,OENA,SELE,MGC=4>         ; 10
001132' 471050  005340
001133' 001100  110000                  MWORD   <JMAP,J=11>                                     ; 11
001134' 000000  000040

                                ;       All done -  set STOP and STORE and write word count to EBUF

001135' 001200  000140                  MWORD   <CONT,SELC,MGC=140>                             ; 12
001136' 000000  004340
001137' 001300  002004                  MWORD   <CONT,SOA,OR,A=5,D=1,OENA,SELE,MGC=4>           ; 13
001140' 431050  005340
001141' 001400  140000                  MWORD   <JMAP,J=14>                                     ; 14
001142' 000000  000040

                                ;       Routine to read a word from CBUS, through formatter into R3
                                ;       (If this routine is started at 101, then the next word from
                                ;       the CBUS is read).

001143' 010000  000200                  MWORD   <ADDR=100,CONT,D=1,SELC,MGC=200>                ; 100
001144' 001000  004340
001145' 010101  030000                  MWORD   <CJP,J=103,D=1,CENA,CCCA>                       ; 101
001146' 001400  000060
001147' 010201  010000                  MWORD   <JMAP,J=101,D=1>                                ; 102
001150' 001000  000040
001151' 010300  000004                  MWORD   <CONT,D=1,SELC,MGC=4>                           ; 103
001152' 001000  004340
001153' 010400  000200                  MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>              ; 104
001154' 732001  402340
001155' 010500  000000                  MWORD   <CRTN,SOB,B=5,PLUS,CRY,D=2>                     ; 105
001156' 302002  400640

                                ;       Routine to read R3 and place in EBUF

```

```
 1103  001157' 020002  012004                 MWORD   <ADDR=200,JMAP,J=201,SOA,A=3,OR,OENA,D=1,SELE,MGC=4>
 1104  001160' 431030  005040
 1105  001161' 020102  010000                 MWORD   <JMAP,J=201>                                ; 201
 1106  001162' 000000  000040
 1107
 1108                                  ;       Initialization
 1109
 1110  001163' 100000  000020                 MWORD   <ADDR=1000,JZ,SELM,MGC=20>                  ; 1000
 1111  001164' 000000  002000
 1112
 1113                                  ;       Initialization step
 1114
 1115  001165' 500050  010100                 MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>        ; 5000
 1116  001166' 000000  004040
 1117  001167' 500150  010020                 MWORD   <JMAP,J=5001,SELM,MGC=20>                   ; 5001
 1118  001170' 000000  002040
 1119  001171' 777777  777777                 -1
```

```
                                       ;#*********************************************************************
                                       ;* Test 46 - CBUS Parity Checking - CBUS Module to CBUS
                                       ;
                                       ; Description:  This test writes a pattern of  floating 1's  over
                                       ;               the CBUS, verifying  with  each  word  transfered
                                       ;               that a parity error has not occurred.
                                       ;
                                       ; Procedure:    KL> Port Clear
                                       ;               KL> Set up a CCW list to transfer 36 words
                                       ;               KL> Start up port
                                       ;               UC> Initialize word count/data pattern
                                       ;               UC> Write 36 words over CBUS to KL10 memory
                                       ;               UC> Assert STOP and STORE and put word count into EBUF
                                       ;               KL> Stop microcode, read EBUF - verify word count
                                       ;               KL> If stopped at wrong location, read R3 to get
                                       ;                   actual data that resulted in parity error.
                                       ;               KL> Verify data transfer completed ok, particularly
                                       ;                   the MB PE bit that should not be set in the channel
                                       ;                   status word.
                                       ;
                                       ; Failure:      ---
                                       ;#*********************************************************************

                                       ; Test data

001172' 254 00 0 00 001200'            TSTC46: JRST    TG46                    ; go start test
001173' 100604  000046                         CBUS!NDMP!NDCB!ZCBUS!46         ; test mask
001174' 001265' 005203'                        T46M,,[ASCIZ ^CBUS Parity Checking - CBUS Module to CBUS^]
001175' 005126' 005130'                        [EXP C17,MLAST!C9],,[EXP MLAST!C7]
001176' 000000  001326'                        TSTC47                          ; failure test table
001177' 777777  777777                         -1                              ;   ...

                                       ; Start test

001200' 201 00 0 00 000000'            TG46:   MOVEI   ZA                      ; get address of module start
001201' 260 17 0 00 001015*                    GO      TRACE                   ; handle trace output
001202' 201 01 0 00 001265'                    MOVEI   1,T46M                  ; set up microcode address
001203' 260 17 0 00 001017*                    GO      TLOAD                   ; load/verify it
001204' 263 17 0 00 000000                     RTN                             ; failed - exit test

                                       ; Initialization

001205' 400 15 0 00 000000             TL46:   SETZ    ERFLG,                  ; clear error flag
001206' 201 06 0 00 001222'                    MOVEI   6,TS46                  ; get test table address
001207' 402 00 0 00 001023*                    SETZM   TSTSUB                  ; initialize subtest number
```

```

                                        ; Loop on test execute table entries

001210' 260 17 0 00 001024*     TA46:   GO      IPACLR                  ; clear port
001211' 260 17 0 00 001025*     TB46:   GO      CEXEC                   ; execute table entry
001212' 254 00 0 00 001221'              JRST   TX46                    ; end of test execute table
001213' 254 00 0 00 001211'              JRST   TB46                    ; keep looping after call
001214' 474 15 0 00 000000               SETO   ERFLG,                  ; error occurred

                                        ; Handle error printouts and scope looping

001215' 027 00 0 00 001236'             SCOPER  MA46                    ; print error message
001216' 254 00 0 00 001205'             JRST    TL46                    ; loop on error
001217' 254 00 0 00 001221'             JRST    TX46                    ; altmode exit
001220' 322 15 0 00 001210'             JUMPE   ERFLG,TA46              ; do next test execute table entry

                                        ; End of test

001221' 263 17 0 00 000000      TX46:   RTN                             ; return

                                        ; Data Test Execute Table, as:  (CMD,parameters)

001222' 040120  000044          TS46:   CTABLE  (CSETRD,120,^D36)       ; set up write (36 words)
001223' 200000  001000                  CTABLE  (CSTART,1000)           ; start up port
001224' 340010  000000                  CTABLE  (CWAIT,10,0)            ; check if completed
001225' 240000  001227'                 CTABLE  (CCALL,TS46CH)          ; check results
001226' 000000  000000                  CTABLE  (CLAST)                 ; end of table

                                        ; Check results

001227' 260 17 0 00 001241'     TS46CH: GO      MA46CH                  ; get additional data
001230' 200 01 0 00 001064*             MOVE    1,CEBUFA                ; get word count
001231' 302 01 0 00 000044              CAIE    1,^D36                  ; correct?
001232' 474 15 0 00 000000              SETO    ERFLG,                  ; no - set error flag
001233' 350 00 0 17 777772              AOS     -6(P)                   ; set up error return
001234' 350 00 0 17 777772              AOS     -6(P)                   ; set up error return
001235' 263 17 0 00 000000              RTN                             ; return

                                        ; Error messages

001236' 140000  005214'         MA46:   MSG!TXNOT![ASCIZ /Data transfer test error detected/]
001237' 240000  001241'                 CALL!TXNOT!MA46CH               ; check other results
001240' 000000000000#                   LAST!CALL!TXALL!CCPNT           ; print test data
```

```

                                          ; Check results - first the word count left in the EBUF

 001241' 322 15 0 00 001252'      MA46CH: JUMPE   ERFLG,MA46C0            ; no error yet? - continue
 001242' 200 01 0 00 001230*              MOVE    1,CEBUFA                ; get word count
 001243' 306 01 0 00 000044               CAIN    1,^D36                  ; correct?
 001244' 254 00 0 00 001252'              JRST    MA46C0                  ; yes - continue
 001245' 037 00 0 00 005141'              TMSGC   <Port word count error:  >
 001246' 037 00 0 00 005147'              TMSGC   <  Correct:   36.>
 001247' 037 00 0 00 005153'              TMSGC   <  Actual:>
 001250' 006 06 0 00 001242*              PDEC    6,CEBUFA                ; print it
 001251' 037 00 0 00 005156'              TMSG    <.>

                                  ; Check for data buffer compare error

 001252' 200 01 0 00 005223'      MA46C0: MOVE    1,[-1]                  ; set to 'read - don't report errors'
 001253' 201 02 0 00 000044               MOVEI   2,^D36                  ; get buffer length
 001254' 260 17 0 00 000000*              GO      BUFCOM                  ; do the compare
 001255' 202 01 0 00 005724'              MOVEM   1,SAVNUM#               ; save number of data compare errors
 001256' 332 00 0 00 000015               SKIPE   ERFLG                   ; error yet?
 001257' 336 00 0 00 000001               SKIPN   1                       ; any words in error?
 001260' 263 17 0 00 000000               RTN                             ; no - return
 001261' 201 01 0 00 000001               MOVEI   1,1                     ; set to 'read - report errors'
 001262' 201 02 0 00 000044               MOVEI   2,^D36                  ; get buffer length
 001263' 260 17 0 00 001254*              GO      BUFCOM                  ; do the compare/print errors
 001264' 263 17 0 00 000000               RTN                             ; done - return
```

```

                               ; Microcode

                               ;       Init word count and data pattern

001265' 000000  010000         T46M:   MWORD   <ADDR=0,JMAP,J=1,SOB,AND,B=5,D=2>                ; 0
001266' 342002  400040
001267' 000100  000000                 MWORD   <CONT,SOA,A=5,B=4,PLUS,CRY,D=2>                  ; 1
001270' 402052  000740

                               ;       Write all the words specified by the CCW list

001271' 000200  000220                 MWORD   <CONT,D=1,SELC,MGC=220>                          ; 2
001272' 001000  004340
001273' 000300  050000                 MWORD   <CJP,J=5,D=1,CENA,CCCA>                          ; 3
001274' 001400  000060
001275' 000400  030000                 MWORD   <JMAP,J=3,D=1>                                   ; 4
001276' 001000  000040
001277' 000500  002040                 MWORD   <CONT,SOA,OR,A=4,D=1,OENA,SELF,MGC=40>           ; 5
001300' 431040  003340
001301' 000600  000002                 MWORD   <CONT,SOB,PLUS,B=5,D=2,CRY,SELC,MGC=2>           ; 6
001302' 302002  404740
001303' 000700  000000                 MWORD   <CONT,SOB,OR,B=4,D=7>                            ; 7
001304' 337002  000340
001305' 001000  120000                 MWORD   <CJP,J=12,SOB,OR,B=4,D=1,CENA,CCFZ>              ; 10
001306' 331402  020060
001307' 001100  030000                 MWORD   <JMAP,J=3>                                       ; 11
001310' 000000  000040

                               ;       All done -  set STOP and STORE

001311' 001200  000140                 MWORD   <CONT,SELC,MGC=140>                              ; 12
001312' 000000  004340
001313' 001300  142004                 MWORD   <JMAP,J=14,SOA,A=5,OR,OENA,D=1,SELE,MGC=4>       ; 13
001314' 431050  005040
001315' 001400  140000                 MWORD   <JMAP,J=14>                                      ; 14
001316' 000000  000040

                               ;       Initialization

001317' 100000  000020                 MWORD   <ADDR=1000,JZ,SELM,MGC=20>                       ; 1000
001320' 000000  002000

                               ;       Initialization step

001321' 500050  010100                 MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>             ; 5000
001322' 000000  004040
001323' 500150  010020                 MWORD   <JMAP,J=5001,SELM,MGC=20>                        ; 5001
001324' 000000  002040
001325' 777777  777777                 -1
```

```
                                        ;#*****************************************************************
                                        ;* Test 47 - CBUS Parity Checking - CBUS to CBUS Module
                                        ;
                                        ; Description:  This test reads a pattern of  floating  1's  over
                                        ;               the CBUS from KL memory, verifying with each word
                                        ;               transfered that a parity error has occurred.
                                        ;
                                        ; Procedure:    KL> Port Clear
                                        ;               KL> Set up a CCW list to transfer 36 words
                                        ;               KL> Start up port
                                        ;               UC> Initialize word count
                                        ;               UC> Read a word from CBUS and put in 2901 R3
                                        ;               UC> Repeat until 'Last Wd' is seen:
                                        ;                      Read a word
                                        ;                      Verify that a parity error has occurred
                                        ;                        If no - put word in error into R3
                                        ;                                - put word count in EBUF
                                        ;                                - abort test
                                        ;                        If yes - do next word
                                        ;               UC> Assert STOP and STORE and put word count into EBUF
                                        ;               KL> Stop microcode, read EBUF - verify word count
                                        ;               KL> If stopped at wrong location, read R3 to get
                                        ;                   actual data that resulted in no parity error.
                                        ;               KL> Verify data transfer completed ok.
                                        ;
                                        ; Failure:      ---
                                        ;#*****************************************************************

                                        ; Test data

 001326' 254 00 0 00 001334'            TSTC47: JRST    TG47                    ; go start test
 001327' 100604  000047                         CBUS!NDMP!NDCB!ZCBUS!47         ; test mask
 001330' 001470' 005115'                        T47M,,[ASCIZ ^CBUS Parity Checking - CBUS to CBUS Module^]
 001331' 005126' 005130'                        [EXP C17,MLAST!C9],,[EXP MLAST!C7]
 001332' 000000  002141'                        TSTC52                          ; failure test table
 001333' 777777  777777                         -1                              ;   ...
```

```

                                ; Start test

001334' 402 00 0 00 005715'     TG47:   SETZM   SAVAPR#                 ; clear APR data
001335' 332 00 0 00 030037              SKIPE   USER                    ; user mode?
                                        JRST    [MOVE   TSTFLG          ; yes - ensure this test does not
                                                 TLO    (TUSER)         ;    get run (but only if not in
                                                 SKIPN  UDEBUG          ;    debug mode)
                                                 MOVEM  TSTFLG
001336' 254 00 0 00 005224'                      JRST   TG47A]
001337' 7 000 24 0 00 000001            CONI    APR,1                   ; get processor flags
001340' 202 01 0 00 005715'             MOVEM   1,SAVAPR                ; save flags
001341' 7 000 20 0 00 065400            CONO    APR,65400               ; prevent MB parity error interrupts
001342' 201 00 0 00 000000'     TG47A:  MOVEI   ZA                      ; get address of module start
001343' 260 17 0 00 001201*             GO      TRACE                   ; handle trace output
001344' 201 01 0 00 001470'             MOVEI   1,T47M                  ; set up microcode address
001345' 260 17 0 00 001203*             GO      TLOAD                   ; load/verify it
001346' 254 00 0 00 001363'             JRST    TX47                    ; failed - exit test

                                ; Initialization

001347' 400 15 0 00 000000      TL47:   SETZ    ERFLG,                  ; clear error flag
001350' 201 06 0 00 001373'             MOVEI   6,TS47                  ; get test table address
001351' 402 00 0 00 001207*             SETZM   TSTSUB                  ; initialize subtest number

                                ; Loop on test execute table entries

001352' 260 17 0 00 001210*     TA47:   GO      IPACLR                  ; clear port
001353' 260 17 0 00 001211*     TB47:   GO      CEXEC                   ; execute table entry
001354' 254 00 0 00 001363'              JRST   TX47                    ; end of test execute table
001355' 254 00 0 00 001353'              JRST   TB47                    ; keep looping after call
001356' 474 15 0 00 000000               SETO   ERFLG,                  ; error occurred

                                ; Handle error printouts and scope looping

001357' 027 00 0 00 001431'             SCOPER  MA47                    ; print error message
001360' 254 00 0 00 001347'             JRST    TL47                    ; loop on error
001361' 254 00 0 00 001363'             JRST    TX47                    ; altmode exit
001362' 322 15 0 00 001352'             JUMPE   ERFLG,TA47              ; do next test execute table entry

                                ; End of test

001363' 336 00 0 00 005715'     TX47:   SKIPN   SAVAPR                  ; need to restore APR flags?
001364' 263 17 0 00 000000              RTN                             ; no - just return
001365' 7 000 20 0 00 027700            CONO    APR,27700               ; clear flags
001366' 135 01 0 00 005231'             LDB     1,[POINT 8,SAVAPR,13]   ; get original flags
001367' 242 01 0 00 000004              LSH     1,4                     ; position properly
001370' 660 01 0 00 100000              TRO     1,100000                ; set enable bit
001371' 7 000 20 0 01 000000            CONO    APR,(1)                 ; set the flags
001372' 263 17 0 00 000000              RTN                             ; exit
```

```
                                        ; Data Test Execute Table, as:  (CMD,parameters)

 001373' 100120  000044                 TS47:   CTABLE  (CSETWR,120,^D36)       ; set up write (36 words)
 001374' 240000  001401'                        CTABLE  (CCALL,TS47DA)          ; set up data patterns correctly
 001375' 200000  001000                         CTABLE  (CSTART,1000)           ; start up port
 001376' 340010  000000                         CTABLE  (CWAIT,10,0)            ; check if completed
 001377' 240000  001416'                        CTABLE  (CCALL,TS47CH)          ; check results
 001400' 000000  000000                         CTABLE  (CLAST)                 ; end of table

                                        ; Set up data pattern - make each word have bad parity

 001401' 200 00 0 00 005232'            TS47DA: MOVE    [BUFF,,BUFF+^D36]       ; build BLT word, save copy of the
 001402' 251 00 0 00 000000#                    BLT     BUFF+^D71               ;   data later in the buffer
 001403' 201 01 0 00 000645*                    MOVEI   1,BUFF                  ; get buffer address
 001404' 200 00 0 01 000000             TS47D0: MOVE    (1)                     ; get data word
 001405' 336 00 0 00 030037                     SKIPN   USER                    ; user mode?
 001406' 7 004 20 0 00 200000                   CONO    PI,200000               ; no - set up for bad parity
 001407' 202 00 0 01 000000                     MOVEM   (1)                     ; write the word back
 001410' 336 00 0 00 030037                     SKIPN   USER                    ; user mode?
 001411' 7 004 20 0 00 000000                   CONO    PI,0                    ; no - reset bad parity flag
 001412' 350 00 0 00 000001                     AOS     1                       ; point to next address
 001413' 307 01 0 00 000000#                    CAIG    1,BUFF+^D35             ; done yet?
 001414' 254 00 0 00 001404'                    JRST    TS47D0                  ; no - loop till done
 001415' 263 17 0 00 000000                     RTN                             ; yes - return

                                        ; Check results

 001416' 200 01 0 00 005233'            TS47CH: MOVE    1,[-^D36,,BUFF]         ; build AOBJN word to rewrite
 001417' 550 02 0 00 000001                     HRRZ    2,1                     ;   each word in the buffer
 001420' 402 00 0 01 000000                     SETZM   (1)                     ;   with correct parity, and
 001421' 253 01 0 00 001417'                    AOBJN   1,.-2                   ;   do so
 001422' 260 17 0 00 001434'                    GO      MA47CH                  ; get additional data
 001423' 200 01 0 00 001250*                    MOVE    1,CEBUFA                ; get word count
 001424' 302 01 0 00 000044                     CAIE    1,^D36                  ; correct?
 001425' 474 15 0 00 000000                     SETO    ERFLG,                  ; no - set error flag
 001426' 350 00 0 17 777772                     AOS     -6(P)                   ; set up error return
 001427' 350 00 0 17 777772                     AOS     -6(P)                   ; set up error return
 001430' 263 17 0 00 000000                     RTN                             ; return

                                        ; Error messages

 001431' 140000  005234'                MA47:   MSG!TXNOT![ASCIZ /Parity error not detected in data transfer/]
 001432' 240000  001434'                        CALL!TXNOT!MA47CH               ; check other results
 001433' 000000000000#                          LAST!CALL!TXALL!CCPNT           ; print test data
```

```

                                   ; Check results - first the word count left in the EBUF

001434' 322 15 0 00 001446'        MA47CH: JUMPE   ERFLG,MA47C0            ; no error yet? - continue
001435' 200 01 0 00 001423*                MOVE    1,CEBUFA                ; get word count
001436' 306 01 0 00 000044                 CAIN    1,^D36                  ; correct?
001437' 254 00 0 00 001434'                JRST    MA47CH0                 ; yes - continue
001440' 331 00 0 00 001435*                SKIPL   CEBUFA
                                           JRST    [TMSGC  <Port word count error:  >
                                                    TMSGC  <  Correct:    36.>
                                                    TMSGC  <  Actual:>
                                                    PDEC   6,CEBUFA                ; print it
                                                    TMSG   <.>
                                                    JRST   MA47C0]
001441' 254 00 0 00 005245'
001442' 037 00 0 00 005253'                TMSGC   <Parity error did not occur on word #>
001443' 210 00 0 00 001440*                MOVN    CEBUFA
001444' 037 15 0 00 000000                 PNTDEC
001445' 037 00 0 00 005156'                TMSG    <.>

                                   ; Then the correct/actual data being read

001446' 400 01 0 00 000000         MA47C0: SETZ    1,                      ; ensure the port is
001447' 260 17 0 00 001103*                GO      LDCSR                   ;   stopped
001450' 201 01 0 00 000200                 MOVEI   1,200                   ; get address to get R3
001451' 242 01 0 00 000001                 LSH     1,1                     ; position correctly
001452' 260 17 0 00 001073*                GO      LDRAR                   ; load RAR register
001453' 201 01 0 00 000010                 MOVEI   1,MPRUN                 ; set up initial CSR data
001454' 260 17 0 00 001447*                GO      LDCSR                   ; write CSR register
001455' 201 01 0 00 200000                 MOVEI   1,TSTEBF                ; set up CSR to read EBUF
001456' 260 17 0 00 001454*                GO      LDCSR                   ; write it
001457' 260 17 0 00 001100*                GO      RDEBUF                  ; get EBUF data (actual)
001460' 202 01 0 00 005722'                MOVEM   1,SAVDA1#               ; save it
001461' 400 01 0 00 000000                 SETZ    1,                      ; ensure the port is
001462' 260 17 0 00 001456*                GO      LDCSR                   ;   stopped
001463' 322 15 0 00 005174'                JUMPE   ERFLG,[RTN]             ; no error yet? - continue
001464' 037 00 0 00 005263'                TMSGC   <Word without parity error:  >
001465' 200 00 0 00 005722'                MOVE    SAVDA1                  ; get data
001466' 037 13 0 00 000000                 PNTHW                           ; print it
001467' 263 17 0 00 000000                 RTN                             ; return
```

```

                                ; Microcode - Read each word

001470' 000000  010000          T47M:   MWORD   <ADDR=0,JMAP,J=1,SOB,AND,B=5,D=2>               ; 0
001471' 342002  400040
001472' 000101  000000                  MWORD   <CJS,J=100,D=1>                                 ; 1
001473' 001000  000020
001474' 000200  040000                  MWORD   <JMAP,J=4,D=1>                                  ; 2
001475' 001000  000040
001476' 000301  010000                  MWORD   <CJS,J=101,D=1>                                 ; 3
001477' 001000  000020
001500' 000400  060000                  MWORD   <CJP,J=6,D=1,CENA,CCCP>                         ; 4
001501' 001400  140060
001502' 000500  100000                  MWORD   <JMAP,J=10,D=1>                                 ; 5
001503' 001000  000040
001504' 000600  130000                  MWORD   <CJP,J=13,D=1,CENA,CCLW>                        ; 6
001505' 001400  170060
001506' 000700  030000                  MWORD   <JMAP,J=3>                                      ; 7
001507' 000000  000040

                                ;       Parity error did not occur - set STOP and STORE and write
                                ;       word count to EBUF

001510' 001000  000140                  MWORD   <CONT,SELC,MGC=140>                             ; 10
001511' 000000  004340
001512' 001100  002004                  MWORD   <CONT,SOA,XNOR,A=5,D=1,OENA,SELE,MGC=4>         ; 11
001513' 471050  005340
001514' 001200  120000                  MWORD   <JMAP,J=12>                                     ; 12
001515' 000000  000040

                                ;       All done -  set STOP and STORE and write word count to EBUF

001516' 001300  000140                  MWORD   <CONT,SELC,MGC=140>                             ; 13
001517' 000000  004340
001520' 001400  002004                  MWORD   <CONT,SOA,OR,A=5,D=1,OENA,SELE,MGC=4>           ; 14
001521' 431050  005340
001522' 001500  150000                  MWORD   <JMAP,J=15>                                     ; 15
001523' 000000  000040

                                ;       Routine to read a word from CBUS, through formatter into R3
                                ;       (If this routine is started at 101, then the next word from
                                ;       the CBUS is read).

001524' 010000  000200                  MWORD   <ADDR=100,CONT,D=1,SELC,MGC=200>                ; 100
001525' 001000  004340
001526' 010101  030000                  MWORD   <CJP,J=103,D=1,CENA,CCCA>                       ; 101
001527' 001400  000060
001530' 010201  010000                  MWORD   <JMAP,J=101,D=1>                                ; 102
001531' 001000  000040
001532' 010300  000004                  MWORD   <CONT,D=1,SELC,MGC=4>                           ; 103
001533' 001000  004340
001534' 010400  000200                  MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>              ; 104
001535' 732001  402340
001536' 010500  000000                  MWORD   <CRTN,SOB,B=5,PLUS,CRY,D=2>                     ; 105
001537' 302002  400640
```

```

                                   ;       Routine to read R3 and place in EBUF

001540' 020002  012004                     MWORD   <ADDR=200,JMAP,J=201,SOA,A=3,OR,OENA,D=1,SELE,MGC=4>
001541' 431030  005040
001542' 020102  010000                     MWORD   <JMAP.J=201>                                   ; 201
001543' 000000  000040

                                   ;       Initialization

001544' 100000  000020                     MWORD   <ADDR=1000,JZ,SELM,MGC=20>                     ; 1000
001545' 000000  002000

                                   ;       Initialization step

001546' 500050  010100                     MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>           ; 5000
001547' 000000  004040
001550' 500150  010020                     MWORD   <JMAP,J=5001,SELM,MGC=20>                      ; 5001
001551' 000000  002040
001552' 777777  777777                     -1
```

```

                                        ;#*******************************************************************
                                        ;* Test 50 - T Field Timing Test
                                        ;
                                        ; Description:  Verify that setting the Time Field  bit  in  each
                                        ;               microword causes the  microsequencer  to  operate
                                        ;               about 33% slower.
                                        ;
                                        ; Procedure:    KL> Port Clear
                                        ;               KL> Load microcode
                                        ;               KL> Start the port at location 0
                                        ;               KL> Wait 10ms then stop the port and read 2901
                                        ;                   register 1 which contains the number of
                                        ;                   instructions executed (each with T field
                                        ;                   not set)
                                        ;               KL> Start the port at location 10
                                        ;               KL> Wait 10ms then stop the port and read 2901
                                        ;                   register 1 which contains the number of
                                        ;                   instructions executed (each with T field
                                        ;                   set)
                                        ;
                                        ;               KL> Compare the instruction counts expecting a
                                        ;                   ratio of 3 to 2 +- 10%
                                        ;
                                        ;               KL> In user mode repeat up to 2 more times if the
                                        ;                   ratio is not correct (it is more difficult to
                                        ;                   exactly time how long the wait is being done).
                                        ;
                                        ; Failure:      ---
                                        ;#*******************************************************************

                                        ; Test data

001553' 254 00 0 00 001560'             TSTC50: JRST    TG50                    ; go start test
001554' 100604  000050                          CBUS!NDMP!NDCB!ZCBUS!50         ; test mask
001555' 001657' 005272'                         T50M,,[ASCIZ ^T Field Timing^]
001556' 005275' 000000                          [EXP MLAST!C1],,0
001557' 777777  777777                          -1                              ; failure test table
```

```

                                ; Start test

001560' 332 00 0 00 030037      TG50:   SKIPE   USER                    ; user mode?
                                        GO      [MOVE   TSTFLG          ; yes - ensure this test does not
                                                 TLO    (TUSER)         ;    get run (but only if not in
                                                 SKIPN  UDEBUG          ;    debug mode)
                                                 MOVEM  TSTFLG
001561' 260 17 0 00 005276'                      RTN]
001562' 201 00 0 00 000000'             MOVEI   ZA                      ; get address of module start
001563' 260 17 0 00 001343*             GO      TRACE                   ; handle trace output
001564' 201 01 0 00 001657'             MOVEI   1,T50M                  ; set up microcode address
001565' 260 17 0 00 001345*             GO      TLOAD                   ; load/verify it
001566' 263 17 0 00 000000              RTN                             ; failed - exit test

                                ; Do the test

001567' 400 15 0 00 000000      TA50:   SETZ    ERFLG,                  ; clear error flag
001570' 400 01 0 00 000000              SETZ    1,                      ; set start address to 0
001571' 260 17 0 00 001616'             GO      T50SS                   ; do a test segment
001572' 202 01 0 00 005716'             MOVEM   1,SAVCN1#               ; save count

001573' 201 01 0 00 000010              MOVEI   1,10                    ; set start address to 10
001574' 260 17 0 00 001616'             GO      T50SS                   ; do a test segment
001575' 202 01 0 00 005717'             MOVEM   1,SAVCN2#               ; save count

                                ; Check results

001576' 200 04 0 00 005716'             MOVE    4,SAVCN1                ; get count without T field set
001577' 221 04 0 00 000012              IMULI   4,^D10                  ; calculate 1.4 times the count
001600' 231 04 0 00 000016              IDIVI   4,^D14                  ;   as the maximum limit
001601' 202 04 0 00 005720'             MOVEM   4,SAVCN3#               ; save it
001602' 317 04 0 00 005717'             CAMG    4,SAVCN2                ; in range?
001603' 474 15 0 00 000000              SETO    ERFLG,                  ; no - set error flag
001604' 200 04 0 00 005716'             MOVE    4,SAVCN1                ; get count without T field set
001605' 221 04 0 00 000012              IMULI   4,^D10                  ; calculate 1.6 times the count
001606' 231 04 0 00 000020              IDIVI   4,^D16                  ;   as the minimum limit
001607' 202 04 0 00 005721'             MOVEM   4,SAVCN4#               ; save it
001610' 311 04 0 00 005717'             CAML    4,SAVCN2                ; in range?
001611' 474 15 0 00 000000              SETO    ERFLG,                  ; no - set error flag

                                ; Handle error printouts and scope looping

001612' 027 00 0 00 001631'             SCOPER  MA50                    ; print error message
001613' 254 00 0 00 001567'             JRST    TA50                    ; loop on error
001614' 254 00 0 00 001615'             JRST    TX50                    ; altmode exit

                                ; End of test

001615' 263 17 0 00 000000      TX50:   RTN                             ; return
```

```

                                         ; Routine to start port, wait 20 msec, then read count

 001616' 260 17 0 00 001352*     T50SS:  GO      IPACLR                  ; do a 'port clear'
 001617' 242 01 0 00 000001              LSH     1,1                     ; position start address
 001620' 260 17 0 00 001452*             GO      LDRAR                   ; set up start address
 001621' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get MPRUN bit
 001622' 260 17 0 00 001462*             GO      LDCSR                   ; start the port
 001623' 201 00 0 00 000024              MOVEI   ^D20                    ; set up for 20 msec
 001624' 260 17 0 00 000000*             GO      ODELAY                  ; do the wait
 001625' 201 01 0 00 200000              MOVEI   1,TSTEBF                ; set up to read EBUF
 001626' 260 17 0 00 001622*             GO      LDCSR                   ; stop the port
 001627' 260 17 0 00 001457*             GO      RDEBUF                  ; read EBUF data
 001630' 263 17 0 00 000000              RTN                             ; return

                                 ; Error messages

 001631' 270000  001632'         MA50:   LAST!CALL!TXALL!MA50PN          ; print timing data

 001632' 037 00 0 00 005303'     MA50PN: TMSGC   <T field set did not slow port by 33%>
 001633' 200 00 0 00 000000*             MOVE    SCOSW                   ; get switches
 001634' 603 00 0 00 000200              TLNE    TXTINH                  ; inhibit text?
 001635' 263 17 0 00 000000              RTN                             ; yes - exit
 001636' 037 00 0 00 005313'             TMSGC   <Count (TFld=0):  >
 001637' 200 00 0 00 005716'             MOVE    SAVCN1                  ; get count
 001640' 037 15 0 00 000000              PNTDEC                          ; print it
 001641' 037 00 0 00 005156'             TMSG    <.>
 001642' 037 00 0 00 005317'             TMSGC   <Count (TFld=1) minimum:  >
 001643' 200 00 0 00 005721'             MOVE    SAVCN4                  ; get count
 001644' 037 15 0 00 000000              PNTDEC                          ; print it
 001645' 037 00 0 00 005156'             TMSG    <.>
 001646' 037 00 0 00 005325'             TMSGC   <Count (TFld=1) maximum:  >
 001647' 200 00 0 00 005720'             MOVE    SAVCN3                  ; get count
 001650' 037 15 0 00 000000              PNTDEC                          ; print it
 001651' 037 00 0 00 005156'             TMSG    <.>
 001652' 037 00 0 00 005333'             TMSGC   <Count (TFld=1) actual:   >
 001653' 200 00 0 00 005717'             MOVE    SAVCN2                  ; get count
 001654' 037 15 0 00 000000              PNTDEC                          ; print it
 001655' 037 00 0 00 005156'             TMSG    <.>
 001656' 263 17 0 00 000000              RTN                             ; return

                                 ; Microcode:

 001657' 000000  010000          T50M:   MWORD   <ADDR=0,JMAP,J=1,SOA,AND,D=2>                        ; 0
 001660' 442000  000040
 001661' 000100  012004                  MWORD   <JMAP,J=1,SOB,PLUS,D=2,CRY,OENA,SELE,MGC=4>          ; 1
 001662' 302000  005440

 001663' 001000  110000                  MWORD   <ADDR=10,JMAP,J=11,SOA,AND,D=2,TIM=1>                   ; 10
 001664' 442000  000050
 001665' 001100  112004                  MWORD   <JMAP,J=11,SOB,PLUS,D=2,CRY,TIM=1,OENA,SELE,MGC=4> ; 11
 001666' 302000  005450
 001667' 777777  777777                  -1
```

```
                                     ;#*********************************************************************
                                     ;* Test 51 - CBUS Store
                                     ;
                                     ; Description:  Verify that the microcode  command  'MPSTORECBUS'
                                     ;               does indeed cause the logout data  to  be  stored
                                     ;               when the command is issued.
                                     ;
                                     ; Procedure:    KL> Port Clear
                                     ;               KL> Set up a CCW list to transfer 1 word
                                     ;               KL> Start up port
                                     ;               UC> Start data transfer
                                     ;               UC> Read a word and assert 'MPSTORECBUS' and 'MPSTOPCBUS'
                                     ;               KL> Read logout data and verify correct
                                     ;
                                     ; Failure:      ---
                                     ;#*********************************************************************

                                     ; Test data

001670' 254 00 0 00 001676'          TSTC51: JRST    TG51                        ; go start test
001671' 100604  000051                       CBUS!NDMP!NDCB!ZCBUS!51             ; test mask
001672' 002116' 005341'                      T51M,,[ASCIZ ^CBUS Store^]
001673' 005275' 000000                       [EXP MLAST!C1],,0
001674' 000000  002141'                      TSTC52                              ; failure test table
001675' 777777  777777                       -1                                  ;   ...

                                     ; Start test

001676' 201 00 0 00 000000'          TG51:   MOVEI   ZA                          ; get address of module start
001677' 260 17 0 00 001563*                  GO      TRACE                       ; handle trace output
001700' 201 01 0 00 002116'                  MOVEI   1,T51M                      ; set up microcode address
001701' 260 17 0 00 001565*                  GO      TLOAD                       ; load/verify it
001702' 263 17 0 00 000000                   RTN                                 ; failed - exit test
001703' 200 00 0 00 005344'                  MOVE    [TS51CD,,TS51TC]            ; set up correct logout data
001704' 251 00 0 00 002011'                  BLT     TS51TC+2

                                     ; Initialization

001705' 400 15 0 00 000000           TL51:   SETZ    ERFLG,                      ; clear error flag
001706' 201 06 0 00 001722'                  MOVEI   6,TS51                      ; get test table address
001707' 402 00 0 00 001351*                  SETZM   TSTSUB                      ; initialize subtest number

                                     ; Loop on test execute table entries

001710' 260 17 0 00 001616*          TA51:   GO      IPACLR                      ; clear port
001711' 260 17 0 00 000615*          TB51:   GO      TEXEC                       ; execute table entry
001712' 254 00 0 00 001721'                   JRST    TX51                       ; end of sstep table
001713' 254 00 0 00 001711'                   JRST    TB51                       ; keep looping after call
001714' 474 15 0 00 000000                    SETO    ERFLG,                     ; error occurred
```

```

                                  ; Handle error printouts and scope looping

001715' 027 00 0 00 002017'             SCOPER  MA51                    ; print error message
001716' 254 00 0 00 001705'             JRST    TL51                    ; loop on error
001717' 254 00 0 00 001721'             JRST    TX51                    ; altmode exit
001720' 322 15 0 00 001710'             JUMPE   ERFLG,TA51              ; do next sstep table entry

                                  ; End of test

001721' 263 17 0 00 000000        TX51:   RTN                             ; return

                                  ; Test Execute Table, as:  (CMD,parameters)

001722' 140000  001727'           TS51:   TTABLE  (TCALL,TS51IN)          ; initialize data transfer
001723' 040000  000000                    TTABLE  (TSTART,0)              ; start up port
001724' 140000  001747'                   TTABLE  (TCALL,TS51CH)          ; check if completed
001725' 340000  000000                    TTABLE  (TEXIT)                 ; exit if error yet
001726' 000000  000000                    TTABLE  (TLAST)                 ; end of table

                                  ; Initialize data transfer

001727' 201 01 0 00 005000        TS51IN: MOVEI   1,5000                  ; get start address
001730' 242 01 0 00 000001                LSH     1,1                     ; position correctly
001731' 260 17 0 00 001620*               GO      LDRAR                   ; load the register
001732' 201 01 0 00 000010                MOVEI   1,MPRUN                 ; get CSR data
001733' 260 17 0 00 001626*               GO      LDCSR                   ; start up the port
001734' 200 01 0 00 005103'               MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
001735' 260 17 0 00 000644*               GO      CHINIT                  ; initialize software
001736' 551 01 0 00 001403*               HRRZI   1,BUFF                  ; buffer address
001737' 201 02 0 00 000001                MOVEI   2,1                     ; word count
001740' 400 03 0 00 000000                SETZ    3,                      ; clear zero fill specifier
001741' 260 17 0 00 000650*               GO      GENCCW                  ; generate a CCW list
001742' 201 00 0 00 123456                MOVEI   123456                  ; initialize buffer
001743' 202 00 0 00 001736*               MOVEM   BUFF                    ;   contents
001744' 400 01 0 00 000000                SETZ    1,                      ; clear CSR data
001745' 260 17 0 00 001733*               GO      LDCSR                   ; stop the port
001746' 263 17 0 00 000000                RTN                             ; return

                                  ; Inspect logout data to see if correct

001747' 201 00 0 00 000001        TS51CH: MOVEI   1                       ; set up a small delay of 1
001750' 260 17 0 00 001624*               GO      ODELAY                  ;   millisecond
001751' 260 17 0 00 000000*               GO      GETLOG                  ; get channel status words
001752' 200 01 0 00 005345'               MOVE    1,[LOGBUF,,TS51TB]      ; transfer to TS51TB
001753' 251 01 0 00 002005'               BLT     1,TS51TB+2
001754' 336 00 0 00 030037                SKIPN   USER                    ; user mode?
001755' 254 00 0 00 001766'               JRST    TS51C0                  ; no - continue
```

```

                                        ; Set up correct data in user mode

001756' 135 00 0 00 005346'                     LDB     [POINT 22,TS51TB+1,35]  ; yes - ignore the CLP
001757' 137 00 0 00 005347'                     DPB     [POINT 22,TS51TC+1,35]  ; save the correct CLP address
001760' 370 00 0 00 000000                      SOS                             ; subtract 1
001761' 137 00 0 00 005350'                     DPB     [POINT 22,TS51TC,35]    ; save the correct ICW address
001762' 201 01 0 00 000000#                     MOVEI   1,BUFF+1                ; get correct buffer address
001763' 336 00 0 00 000000*                     SKIPN   UDEBUG                  ; debug mode?
001764' 257 01 0 01 000000                      MAP     1,(1)                   ; no - map to physical
001765' 137 01 0 00 005351'                     DPB     1,[POINT 22,TS51TC+2,35]; save it in correct location

                                        ; Now do the compare

001766' 200 01 0 00 002003'             TS51C0: MOVE    1,TS51TB                ; get ICWA
001767' 312 01 0 00 002007'                     CAME    1,TS51TC                ; correct?
001770' 474 15 0 00 000000                      SETO    ERFLG,                  ; no - flag error
001771' 200 01 0 00 002004'                     MOVE    1,TS51TB+1              ; get status word 1
001772' 312 01 0 00 002010'                     CAME    1,TS51TC+1              ; correct?
001773' 474 15 0 00 000000                      SETO    ERFLG,                  ; no - flag error
001774' 200 01 0 00 002005'                     MOVE    1,TS51TB+2              ; get status word 2
001775' 312 01 0 00 002011'                     CAME    1,TS51TC+2              ; correct?
001776' 474 15 0 00 000000                      SETO    ERFLG,                  ; no - flag error
001777' 336 00 0 00 000015                      SKIPN   ERFLG                   ; error flag set?
002000' 350 00 0 17 777772                      AOS     -6(P)                   ; no - set up good return
002001' 350 00 0 17 777772                      AOS     -6(P)                   ; set up return
002002' 263 17 0 00 000000                      RTN                             ; return

002003'                                 TS51TB: BLOCK   4                       ; actual data
002007'                                 TS51TC: BLOCK   4                       ; correct data

002013' 200000  000000*                 TS51CD: 200000,,CBUF                    ; initialization data
002014' 500000  000000#                         500000,,CBUF+1
002015' 600000  000000#                         600000,,BUFF+1
002016' 000000  000000                          0

                                        ; Error messages

002017' 140000  005352'                 MA51:   MSG!TXNOT![ASCIZ /'MPSTORECBUS' did not cause logout data to be updated/]
002020' 270000  002021'                         LAST!CALL!TXALL!MA51PN  ; print logout data

002021' 200 01 0 00 001633*             MA51PN: MOVE    1,SCOSW                 ; get switches
002022' 603 01 0 00 000200                      TLNE    1,TXTINH                ; full printout?
002023' 037 00 0 00 005365'                     TMSGC   <Logout (C):>           ; no
002024' 607 01 0 00 000200                      TLNN    1,TXTINH                ; full printout?
002025' 037 00 0 00 005370'                     TMSGC   <Logout Data (Correct):>; yes
002026' 201 03 0 00 002007'                     MOVEI   3,TS51TC                ; get address of correct data
002027' 260 17 0 00 002037'                     GO      MA51LG                  ; print it
```

```
002030' 603 01 0 00 000200              TLNE    1,TXTINH                ; full printout?
002031' 037 00 0 00 005375'             TMSGC   <Logout (A):>           ; no
002032' 607 01 0 00 000200              TLNN    1,TXTINH                ; full printout?
002033' 037 00 0 00 005400'             TMSGC   <Logout Data (Actual):> ; yes
002034' 201 03 0 00 002003'             MOVEI   3,TS51TB                ; get address of actual data
002035' 260 17 0 00 002037'             GO      MA51LG                  ; print it
002036' 263 17 0 00 000000              RTN                             ; return

                                ; Print logout data

002037' 607 01 0 00 000200      MA51LG: TLNN    1,TXTINH                ; full printout
002040' 254 00 0 00 002053'             JRST    MA51LA                  ; yes - continue
002041' 037 00 0 00 005405'             TMSGC   < ICW:  >
002042' 200 00 0 03 000000              MOVE    (3)                     ; get initial CW
002043' 037 13 0 00 000000              PNTHW                           ; print it
002044' 037 00 0 00 005407'             TMSGC   < SW1:  >
002045' 200 00 0 03 000001              MOVE    1(3)                    ; get status word 1
002046' 037 13 0 00 000000              PNTHW                           ; print it
002047' 037 00 0 00 005411'             TMSGC   < SW2:  >
002050' 200 00 0 03 000002              MOVE    2(3)                    ; get last CCW
002051' 037 13 0 00 000000              PNTHW                           ; print it
002052' 263 17 0 00 000000              RTN                             ; done - return

002053' 037 00 0 00 005413'     MA51LA: TMSGC   <  ICW:  >
002054' 200 00 0 03 000000              MOVE    (3)                     ; get initial CW
002055' 037 13 0 00 000000              PNTHW                           ; print it
002056' 037 00 0 00 005416'             TMSGC   <  SW1: >
002057' 200 02 0 03 000001              MOVE    2,1(3)                  ; get status word
002060' 603 02 0 00 200000              TLNE    2,(1B1)                 ; MEM PE?
002061' 037 00 0 00 005420'             TMSG    < MEM.PE>               ; yes - print it
002062' 607 02 0 00 100000              TLNN    2,(1B2)                 ; ADR PE?
002063' 037 00 0 00 005422'             TMSG    < ADR.PE>               ; yes - print it
002064' 607 02 0 00 040000              TLNN    2,(1B3)                 ; WCO?
002065' 037 00 0 00 005424'             TMSG    < WCO>                  ; yes - print it
002066' 603 02 0 00 020000              TLNE    2,(1B4)                 ; NXM?
002067' 037 00 0 00 005425'             TMSG    < NXM>                  ; yes - print it
002070' 603 02 0 00 000400              TLNE    2,(1B9)                 ; LAST XFER ERROR?
002071' 037 00 0 00 005426'             TMSG    < LST.XFR.ERR>          ; yes - print it
002072' 603 02 0 00 000200              TLNE    2,(1B10)                ; DEVICE ERROR?
002073' 037 00 0 00 005431'             TMSG    < DEV.ERR>              ; yes - print it
002074' 603 02 0 00 000100              TLNE    2,(1B11)                ; LONG WC?
002075' 037 00 0 00 005433'             TMSG    < LG.WC>                ; yes - print it
002076' 603 02 0 00 000040              TLNE    2,(1B12)                ; SHORT WC?
002077' 037 00 0 00 005435'             TMSG    < SH.WC>                ; yes - print it
002100' 603 02 0 00 000020              TLNE    2,(1B13)                ; OVERRUN?
002101' 037 00 0 00 005437'             TMSG    < OVR>                  ; yes - print it
002102' 037 00 0 00 005440'             TMSG    <  CLP points to: >
002103' 135 00 0 00 005444'             LDB     [POINT 22,2,35]         ; fetch the actual clp address
002104' 037 16 0 00 000003              PNTOCS                          ; print in octal
```

```
  1867  002105' 037 00 0 00 005445'               TMSGC   <  SW2 (Last Updated CCW): >
  1868  002106' 037 00 0 00 005453'               TMSG    <   WC = >
  1869  002107' 200 02 0 03 000002                MOVE    2,2(3)                  ; fetch last CCW
  1870  002110' 135 00 0 00 005455'               LDB     [POINT  11,2,13]        ; gets word count
  1871  002111' 037 15 0 00 000000                PNTDEC                          ; print it in decimal
  1872  002112' 037 00 0 00 005456'               TMSG    <.  ADR = >
  1873  002113' 135 00 0 00 005444'               LDB     [POINT  22,2,35]        ; gets address
  1874  002114' 037 16 0 00 000003                PNTOCS                          ; print it in octal
  1875  002115' 263 17 0 00 000000                RTN                             ; and exit
  1876
  1877                                    ; Microcode
  1878
  1879  002116' 000000  010200            T51M:   MWORD   <ADDR=0,JMAP,J=1,SELC,MGC=200>          ; 0
  1880  002117' 000000  004040
  1881  002120' 000100  030000                    MWORD   <CJP,J=3,D=1,CENA,CCCA>                 ; 1
  1882  002121' 001400  000060
  1883  002122' 000200  010000                    MWORD   <JMAP,J=1,D=1>                          ; 2
  1884  002123' 001000  000040
  1885  002124' 000300  000004                    MWORD   <CONT,D=1,SELC,MGC=4>                   ; 3
  1886  002125' 001000  004340
  1887  002126' 000400  000200                    MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>      ; 4
  1888  002127' 732001  402340
  1889  002130' 000500  000140                    MWORD   <CONT,SELC,MGC=140>                     ; 5
  1890  002131' 000000  004340
  1891  002132' 000600  060000                    MWORD   <JMAP,J=6>                              ; 6
  1892  002133' 000000  000040
  1893
  1894                                    ;       Initialization steps
  1895
  1896  002134' 500050  010100                    MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>    ; 5000
  1897  002135' 000000  004040
  1898  002136' 500150  010020                    MWORD   <JMAP,J=5001,SELM,MGC=20>               ; 5001
  1899  002137' 000000  002040
  1900  002140' 777777  777777                    -1
```

```

                                      ;#*********************************************************************
                                      ;* Test 52 - CBUS Input Buffer
                                      ;
                                      ; Description:  Verify that the CBUS Input Buffer is cleared by a
                                      ;               'Port Clear'.
                                      ;
                                      ; Procedure:    KL> Port Clear
                                      ;               KL> Load microcode
                                      ;
                                      ;               KL> Single step 3 times to read the formatter into
                                      ;                   the EBUF.  Verify that the EBUF contains 0's.
                                      ;
                                      ; Failure:      ---
                                      ;#*********************************************************************

                                      ; Test data

002141' 254 00 0 00 002146'           TSTC52: JRST    TG52                    ; go start test
002142' 100604  000052                        CBUS!NDMP!NDCB!ZCBUS!52         ; test mask
002143' 002200' 005460'                       T52M,,[ASCIZ ^CBUS Input Buffer^]
002144' 005464' 005466'                       [EXP C14,MLAST!C4],,[EXP C20,MLAST!C7]
002145' 777777  777777                        -1                              ; failure test table

                                      ; Start test

002146' 201 00 0 00 000000'           TG52:   MOVEI   ZA                      ; get address of module start
002147' 260 17 0 00 001677*                   GO      TRACE                   ; handle trace output
002150' 201 01 0 00 002200'                   MOVEI   1,T52M                  ; set up microcode address
002151' 260 17 0 00 001701*                   GO      TLOAD                   ; load/verify it
002152' 263 17 0 00 000000                    RTN                             ; failed - exit test

                                      ; Initialization

002153' 400 15 0 00 000000            TL52:   SETZ    ERFLG,                  ; clear error flag
002154' 260 17 0 00 001710*                   GO      IPACLR                  ; clear port
002155' 402 00 0 00 001707*                   SETZM   TSTSUB                  ; initialize subtest number
002156' 201 06 0 00 002170'                   MOVEI   6,TS52                  ; get sstep table address

                                      ; Loop on single step table entries

002157' 260 17 0 00 000523*           TA52:   GO      BEXEC                   ; execute table entry
002160' 254 00 0 00 002167'                   JRST    TX52                    ; end of sstep table
002161' 254 00 0 00 002157'                   JRST    TA52                    ; keep looping after call
002162' 474 15 0 00 000000                    SETO    ERFLG,                  ; error - EBUF data incorrect
```

```

                                           ; Handle error printouts and scope looping

002163' 027 00 0 00 002176'                        SCOPER  MA52                    ; print error message
002164' 254 00 0 00 002153'                        JRST    TL52                    ; loop on error
002165' 254 00 0 00 002167'                        JRST    TX52                    ; altmode exit
002166' 322 15 0 00 002157'                        JUMPE   ERFLG,TA52              ; do next sstep table entry

                                           ; End of test

002167' 263 17 0 00 000000                 TX52:   RTN                             ; return

                                           ; Sstep table, with EBUF data, as:  (CMD,#Steps,Start addr,End Addr,EBUF data)

002170' 100200  030004                     TS52:   ATABLE  (SSSTRT,2,3,4,0)
002171' 000000  000000
002172' 000000000000#                              ATABLE  (SSCALL,IPACLR)
002173' 100300  000002                             ATABLE  (SSSTRT,3,0,2,0)
002174' 000000  000000
002175' 000000  000000                             ATABLE  (SSLAST)

                                           ; Error messages

002176'                                    MA52:   MSG!TXNOT![ASCIZ /CBUS Input Buffer (read into EBUF) not cleared by a 'PORT CLEAR'/]
002176' 140000  005470'
002177' 000000000000#                              LAST!CALL!TXALL!BBPNT           ; print sequencer data

                                           ; Microcode:

002200' 000000  010004                     T52M:   MWORD   <ADDR=0,JMAP,J=1,D=1,SELC,MGC=4>                ; 0
002201' 001000  004040                             MWORD   <CONT,SD0,OR,D=2,SELM,MGC=200>                  ; 1
002202' 000100  000200
002203' 732000  002340                             MWORD   <JMAP,J=2,SOA,OR,D=1,OENA,SELE,MGC=4>           ; 2
002204' 000200  022004
002205' 431000  005040

002206' 000300  040100                             MWORD   <JMAP,J=4,SELC,MGC=100>                         ; 3
002207' 000000  004040
002210' 000400  042004                             MWORD   <JMAP,J=4,SOA,D=1,AND,OENA,SELE,MGC=4>          ; 4
002211' 441000  005040
002212' 777777  777777                             -1
```

```
                                        ;#*******************************************************************
                                        ;* Test 53 - Parity Predictor - CBIN to Fmtr/MPINDSTCOMP=0
                                        ;
                                        ; Description:  Verify that the parity predictor works  correctly
                                        ;               when moving data from CBIN to the formatter  with
                                        ;               'MPINDSTCOMP'  not  asserted.   This verifies one
                                        ;               half of the predictor.
                                        ;
                                        ; Procedure:    KL> Port Clear
                                        ;               KL> Set up a CCW list to transfer 60 words
                                        ;               KL> Start up port
                                        ;
                                        ;               UC> Read word from CBUS into CBIN
                                        ;               UC> Load formatter with word from CBIN
                                        ;               UC> Test leftmost bit of data word - this should
                                        ;                   be the setting of 'CCMVRPARCHK'.
                                        ;               UC> if correct - set bit 12 (DISABLE Complete)
                                        ;               UC> if not correct - set bit 13 (ENABLE Complete)
                                        ;               Read CSR and determine test disposition
                                        ;
                                        ;               Repeat the above steps for each of the 60 data
                                        ;               words.
                                        ;
                                        ; Failure:      ---
                                        ;#*******************************************************************

                                        ; Test data

 002213' 254 00 0 00 002225'            TSTC53: JRST    TG53                    ; go start test
 002214' 100604  000053                         CBUS!NDMP!NDCB!ZCBUS!53         ; test mask
 002215' 002445' 005505'                        T53M,,[ASCIZ ^Parity Predictor - CBIN to Fmtr/MPINDSTCOMP=0^]
 002216' 005517' 005520'                        [EXP MLAST!C11],,[EXP C12,C14,MLAST!C21]
 002217' 000000  002600'                        TSTC54                          ; failure test table
 002220' 000000  003150'                        TSTC55                          ;   ...
 002221' 000000  003471'                        TSTC56
 002222' 000000  004013'                        TSTC57
 002223' 000000  004352'                        TSTC60
 002224' 777777  777777                         -1

                                        ; Start test

 002225' 201 00 0 00 000000'            TG53:   MOVEI   ZA                      ; get address of module start
 002226' 260 17 0 00 002147*                    GO      TRACE                   ; handle trace output
 002227' 201 01 0 00 002445'                    MOVEI   1,T53M                  ; set up microcode address
 002230' 260 17 0 00 002151*                    GO      TLOAD                   ; load/verify it
 002231' 263 17 0 00 000000                     RTN                             ; failed - exit test
```

```
                                        ; Initialization

002232' 400 15 0 00 000000              TL53:   SETZ    ERFLG,                  ; clear error flag
002233' 260 17 0 00 002154*                     GO      IPACLR                  ; clear port
002234' 402 00 0 00 002155*                     SETZM   TSTSUB                  ; initialize subtest number
002235' 201 06 0 00 002267'                     MOVEI   6,TS53                  ; get test table address

                                        ; Loop on test execute table entries

002236' 260 17 0 00 001711*             TA53:   GO      TEXEC                   ; execute table entry
002237' 254 00 0 00 002266'                      JRST   TX53                    ; end of test execute table
002240' 254 00 0 00 002236'                      JRST   TA53                    ; keep looping after call
002241' 474 15 0 00 000000                       SETO   ERFLG,                  ; error occurred

                                        ; Get miscellaneous data

002242' 201 01 0 00 000300                      MOVEI   1,300                   ; get start address
002243' 242 01 0 00 000001                      LSH     1,1                     ; position correctly
002244' 260 17 0 00 001731*                     GO      LDRAR                   ; load the register
002245' 201 01 0 00 220010                      MOVEI   1,TSTEBF!MPRUN!SINCYC   ; get start data
002246' 260 17 0 00 001745*                     GO      LDCSR                   ; start up the port
002247' 260 17 0 00 001627*                     GO      RDEBUF                  ; read EBUF data
002250' 202 01 0 00 005730'                     MOVEM   1,SAVSEG                ; save segment number
002251' 202 01 0 00 002234*                     MOVEM   1,TSTSUB                ; set up subtest number
002252' 350 00 0 00 002251*                     AOS     TSTSUB                  ;   ...

002253' 201 01 0 00 000200                      MOVEI   1,200                   ; get start address
002254' 242 01 0 00 000001                      LSH     1,1                     ; position correctly
002255' 260 17 0 00 002244*                     GO      LDRAR                   ; load the register
002256' 201 01 0 00 220010                      MOVEI   1,TSTEBF!MPRUN!SINCYC   ; get start data
002257' 260 17 0 00 002246*                     GO      LDCSR                   ; start up the port
002260' 260 17 0 00 002247*                     GO      RDEBUF                  ; read EBUF data
002261' 202 01 0 00 005723'                     MOVEM   1,SAVDAT#               ; save data read

                                        ; Handle error printouts and scope looping

002262' 027 00 0 00 002430'                     SCOPER  MA53                    ; print error message
002263' 254 00 0 00 002232'                     JRST    TL53                    ; loop on error
002264' 254 00 0 00 002266'                     JRST    TX53                    ; altmode exit
002265' 322 15 0 00 002236'                     JUMPE   ERFLG,TA53              ; do next test execute table entry

                                        ; End of test

002266' 263 17 0 00 000000              TX53:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

002267' 140000  002273'                 TS53:   TTABLE  (TCALL,TS53DA)          ; set up data transfer
002270' 040000  001000                          TTABLE  (TSTART,1000)           ; start up port
002271' 240002  000000                          TTABLE  (TCHECK,2)              ; check if completed
002272' 000000  000000                          TTABLE  (TLAST)                 ; end of table
```

```
                                         ; Set up data transfer

002273' 400 01 0 00 000000               TS53DA: SETZ    1,                  ; clear CSR data
002274' 260 17 0 00 002257*                      GO      LDCSR               ; write CSR register
002275' 201 01 0 00 005000                       MOVEI   1,5000              ; get start address
002276' 242 01 0 00 000001                       LSH     1,1                 ; position correctly
002277' 260 17 0 00 002255*                      GO      LDRAR               ; load the register
002300' 201 01 0 00 000010                       MOVEI   1,MPRUN             ; get start data
002301' 260 17 0 00 002274*                      GO      LDCSR               ; start up the port
002302' 201 00 0 00 000001                       MOVEI   1                   ; wait a bit (1 msec)
002303' 260 17 0 00 001750*                      GO      ODELAY
002304' 400 01 0 00 000000                       SETZ    1,                  ; clear CSR data
002305' 260 17 0 00 002301*                      GO      LDCSR               ; stop the port
002306' 200 01 0 00 005103'                      MOVE    1,[CBUFSZ,,CBUF]    ; get pointer to control buffer
002307' 260 17 0 00 001735*                      GO      CHINIT              ; initialize software
002310' 551 01 0 00 001743*                      HRRZI   1,BUFF              ; buffer address
002311' 201 02 0 00 000110                       MOVEI   2,^D72              ; word count
002312' 400 03 0 00 000000                       SETZ    3,                  ; clear zero fill specifier
002313' 260 17 0 00 001741*                      GO      GENCCW              ; generate a CCW list
002314' 200 01 0 00 005523'                      MOVE    1,[TS53BU,,BUFF]    ; copy the data for this test
002315' 251 01 0 00 000000#                      BLT     1,BUFF+^D71         ;   into the data buffer
002316' 476 00 0 00 005730'                      SETOM   SAVSEG#             ; initialize test segment
002317' 263 17 0 00 000000                       RTN                         ; return

                                         ; Data

002320' 000000  000000                   TS53BU: 0
002321' 000000  000000                           0
002322' 400001  000020                           400001,,20
002323' 000000  000020                           20
002324' 400003  000000                           400003,,0
002325' 000001  000000                           1,,0
002326' 000001  000020                           1,,20
002327' 000001  000020                           1,,20
002330' 000000  000000                           0
002331' 400001  000020                           400001,,20
002332' 000001  000000                           1,,0
002333' 000001  000020                           1,,20

002334' 400001  000001                           400001,,1
002335' 000000  000001                           1
002336' 000000  000021                           21
002337' 000000  000021                           21
002340' 000001  000001                           1,,1
002341' 000001  000001                           1,,1
002342' 400003  000021                           400003,,21
002343' 000001  000021                           1,,21
002344' 400001  000001                           400001,,1
002345' 400001  000021                           400001,,21
002346' 400003  000001                           400003,,1
002347' 000001  000021                           1,,21
```

```

002350' 400001  000002                  400001,,2
002351' 000000  000002                  2
002352' 000000  000022                  22
002353' 000000  000022                  22
002354' 000001  000002                  1,,2
002355' 000001  000002                  1,,2
002356' 400003  000022                  400003,,22
002357' 000001  000022                  1,,22
002360' 400001  000002                  400001,,2
002361' 400001  000022                  400001,,22
002362' 400003  000002                  400003,,2
002363' 000001  000022                  1,,22

002364' 400001  000004                  400001,,4
002365' 000000  000004                  4
002366' 000000  000024                  24
002367' 000000  000024                  24
002370' 000001  000004                  1,,4
002371' 000001  000004                  1,,4
002372' 400003  000024                  400003,,24
002373' 000001  000024                  1,,24
002374' 400001  000004                  400001,,4
002375' 400001  000024                  400001,,24
002376' 400003  000004                  400003,,4
002377' 000001  000024                  1,,24

002400' 400001  000010                  400001,,10
002401' 000000  000010                  10
002402' 000000  000030                  30
002403' 000000  000030                  30
002404' 000001  000010                  1,,10
002405' 000001  000010                  1,,10
002406' 400003  000030                  400003,,30
002407' 000001  000030                  1,,30
002410' 400001  000010                  400001,,10
002411' 400001  000030                  400001,,30
002412' 400003  000010                  400003,,10
002413' 000001  000030                  1,,30

002414' 000000  000003                  3
002415' 000000  000003                  3
002416' 400001  000023                  400001,,23
002417' 000000  000023                  23
002420' 400003  000003                  400003,,3
002421' 000001  000003                  1,,3
002422' 000001  000023                  1,,23
002423' 000001  000023                  1,,23
002424' 000000  000003                  3
002425' 400001  000023                  400001,,23
002426' 000001  000003                  1,,3
002427' 000001  000023                  1,,23
```

```


                                    ; Error messages
002430' 140000  005524'             MA53:   MSG!TXNOT![ASCIZ /Parity Predictor produced incorrect result:/]
002431' 240000  002433'                     CALL!TXNOT!MA53PN               ; print result
002432' 000000000000#                       LAST!CALL!TXALL!TTPNT           ; print test data

002433' 200 01 0 00 005730'         MA53PN: MOVE    1,SAVSEG                ; get test segment
002434' 037 00 0 00 005535'                 TMSGC   <  Data in CBIN:  >
002435' 200 00 0 01 002320'                 MOVE    TS53BU(1)               ; get data word
002436' 037 13 0 00 000000                  PNTHW                           ; print it
002437' 200 01 0 01 002320'                 MOVE    1,TS53BU(1)             ; get data word
002440' 607 01 0 00 400000                  TLNN    1,400000                ; should be cleared?
002441' 037 00 0 00 005541'                 TMSGC   <  'CCMVPARCHK' should not be asserted, but is>
002442' 603 01 0 00 400000                  TLNE    1,400000                ; should be set?
002443' 037 00 0 00 005553'                 TMSGC   <  'CCMVPARCHK' should be asserted, but isn't>
002444' 263 17 0 00 000000                  RTN                             ; return

                                    ; Microcode

                                    ;       Read 1st word

002445' 100000  000030              T53M:   MWORD   <ADDR=1000,JZ,SELM,MGC=30>                      ; 1000
002446' 000000  002000
002447' 000000  100000                      MWORD   <ADDR=0,LDCT,J=10>                              ; 0
002450' 000000  000300
002451' 000100  010000                      MWORD   <RPCT,J=1>                                      ; 1
002452' 000000  000220
002453' 000200  000000                      MWORD   <CONT,SOB,AND,B=5,D=2>                          ; 2
002454' 342002  400340
002455' 000301  000000                      MWORD   <CJS,J=100,D=1>                                 ; 3
002456' 001000  000020
002457' 000400  060000                      MWORD   <JMAP,J=6>                                      ; 4
002460' 000000  000040

                                    ;       Read 2nd through nth words

002461' 000501  030000                      MWORD   <CJS,J=103,D=1>                                 ; 5
002462' 001000  000020
002463' 000600  110000                      MWORD   <CJP,J=11,SOA,OR,A=3,D=1,CENA,CCMB>             ; 6
002464' 431430  120060
002465' 000752  000000                      MWORD   <CJP,J=5200,D=1,CENA,CCMP>                      ; 7
002466' 001400  130060
002467' 001000  130000                      MWORD   <JMAP,J=13,D=1>                                 ; 10
002470' 001000  000040
002471' 001100  130000                      MWORD   <CJP,J=13,D=1,CENA,CCMP>                        ; 11
002472' 001400  130060
002473' 001252  000000                      MWORD   <JMAP,J=5200,D=1>                               ; 12
002474' 001000  000040
002475' 001300  150000                      MWORD   <CJP,J=15,SOA,PLUS,A=5,B=5,D=2,CRY,CENA,CCLW>   ; 13
002476' 402452  570460
002477' 001400  050000                      MWORD   <JMAP,J=5>                                      ; 14
002500' 000000  000040

                                    ;       All done -  set STOP and STORE
```

```

002501' 001551  000140          MWORD   <JMAP,J=5100,SELC,MGC=140>                      ; 15
002502' 000000  004040

                         ;      Routine to read a word from CBUS, through formatter into R3
                         ;      (If this routine is started at 101, then the next word from
                         ;      the CBUS is read).

002503' 010000  000200          MWORD   <ADDR=100,CONT,D=1,SELC,MGC=200>                ; 100
002504' 001000  004340
002505' 010100  100000          MWORD   <LDCT,J=10,D=1>                                 ; 101
002506' 001000  000300
002507' 010201  020000          MWORD   <RPCT,J=102,D=1>                                ; 102
002510' 001000  000220
002511' 010301  050000          MWORD   <CJP,J=105,D=1,CENA,CCCA>                       ; 103
002512' 001400  000060
002513' 010401  030000          MWORD   <JMAP,J=103,D=1>                                ; 104
002514' 001000  000040
002515' 010500  000004          MWORD   <CONT,D=1,SELC,MGC=4>                           ; 105
002516' 001000  004340
002517' 010600  000200          MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>              ; 106
002520' 732001  402340
002521' 010700  000000          MWORD   <CRTN,D=1>                                      ; 107
002522' 001000  000240

                         ;      Routine to read R3 and place in EBUF

002523' 020002  002004          MWORD   <ADDR=200,JMAP,J=200,SOA,A=3,OR,OENA,D=1,SELE,MGC=4>
002524' 431030  005040

                         ;      Routine to read R5 and place in EBUF

002525' 030003  002004          MWORD   <ADDR=300,JMAP,J=300,SOA,A=5,OR,OENA,D=1,SELE,MGC=4>
002526' 431050  005040

                         ;      Initialization step

002527' 500050  010100          MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>            ; 5000
002530' 000000  004040
002531' 500150  010020          MWORD   <JMAP,J=5001,SELM,MGC=20>                       ; 5001
002532' 000000  002040

                         ; Subroutine - Passed - Write 1 to bit 12 of CSR register

002533' 510000  000000          MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>        ; 5100
002534' 742001  000340
002535' 510100  000000          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5101
002536' 302001  000740
002537' 510200  260000          MWORD   <LDCT,J=26>                             ; 5102
002540' 000000  000300
002541' 510351  030000          MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>      ; 5103
002542' 102021  000220
002543' 510400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5104
002544' 431020  005340
002545' 510551  070040          MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>  ; 5105
```

```
 2301  002546' 001400  015060
 2302  002547' 510651  050000            MWORD   <JMAP,J=5105,D=1>                           ; 5106
 2303  002550' 001000  000040
 2304  002551' 510700  000200            MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>          ; 5107
 2305  002552' 431010  005340
 2306  002553' 511051  100000            MWORD   <JMAP,J=5110,D=1>                           ; 5110
 2307  002554' 001000  000040
 2308
 2309                            ; Subroutine - Failed - Write 1 to bit 13 of CSR register
 2310
 2311  002555' 520000  000000            MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>            ; 5200
 2312  002556' 742001  000340
 2313  002557' 520100  000000            MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                 ; 5201
 2314  002560' 302001  000740
 2315  002561' 520200  250000            MWORD   <LDCT,J=25>                                 ; 5202
 2316  002562' 000000  000300
 2317  002563' 520352  030000            MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>          ; 5203
 2318  002564' 102021  000220
 2319  002565' 520400  002004            MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>       ; 5204
 2320  002566' 431020  005340
 2321  002567' 520552  070040            MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>      ; 5205
 2322  002570' 001400  015060
 2323  002571' 520652  050000            MWORD   <JMAP,J=5205,D=1>                           ; 5206
 2324  002572' 001000  000040
 2325  002573' 520700  000200            MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>          ; 5207
 2326  002574' 431010  005340
 2327  002575' 521052  100000            MWORD   <JMAP,J=5210,D=1>                           ; 5210
 2328  002576' 001000  000040
 2329  002577' 777777  777777            -1
```

```
                                   ;#*******************************************************************
                                   ;* Test 54 - Parity Predictor - CBIN to Fmtr/MPINDSTCOMP=1
                                   ;
                                   ; Description:  Verify that the parity predictor works  correctly
                                   ;               when moving data from CBIN to the formatter  with
                                   ;               'MPINDSTCOMP' asserted.  This verifies  one  half
                                   ;               of the predictor.
                                   ;
                                   ; Procedure:    KL> Port Clear
                                   ;               KL> Set up a CCW list to transfer 16 words
                                   ;               KL> Start up port
                                   ;
                                   ;               UC> Read word from CBUS into CBIN
                                   ;               UC> Load formatter with word from CBIN
                                   ;               UC> Test leftmost bit of data word - this should
                                   ;                   be the setting of 'CCMVRPARCHK'.
                                   ;               UC> if correct - set bit 12 (DISABLE Complete)
                                   ;               UC> if not correct - set bit 13 (ENABLE Complete)
                                   ;               Read CSR and determine test disposition
                                   ;
                                   ;               Repeat the above steps for each of the 16 data
                                   ;               words.
                                   ;
                                   ; Failure:      ---
                                   ;#*******************************************************************

                                   ; Test data

002600' 254 00 0 00 002611'        TSTC54: JRST    TG54                    ; go start test
002601' 100604  000054                     CBUS!NDMP!NDCB!ZCBUS!54         ; test mask
002602' 003015' 005565'                    T54M,,[ASCIZ ^Parity Predictor - CBIN to Fmtr/MPINDSTCOMP=1^]
002603' 005517' 005520'                    [EXP MLAST!C11],,[EXP C12,C14,MLAST!C21]
002604' 000000  003150'                    TSTC55                          ; failure test table
002605' 000000  003471'                    TSTC56                          ;   ...
002606' 000000  004013'                    TSTC57
002607' 000000  004352'                    TSTC60
002610' 777777  777777                     -1

                                   ; Start test

002611' 201 00 0 00 000000'        TG54:   MOVEI   ZA                      ; get address of module start
002612' 260 17 0 00 002226*                GO      TRACE                   ; handle trace output
002613' 201 01 0 00 003015'                MOVEI   1,T54M                  ; set up microcode address
002614' 260 17 0 00 002230*                GO      TLOAD                   ; load/verify it
002615' 263 17 0 00 000000                 RTN                             ; failed - exit test
```

```


                                        ; Initialization

002616' 400 15 0 00 000000              TL54:   SETZ    ERFLG,                  ; clear error flag
002617' 260 17 0 00 002233*                     GO      IPACLR                  ; clear port
002620' 402 00 0 00 002252*                     SETZM   TSTSUB                  ; initialize subtest number
002621' 201 06 0 00 002653'                     MOVEI   6,TS54                  ; get test table address

                                        ; Loop on test execute table entries

002622' 260 17 0 00 002236*             TA54:   GO      TEXEC                   ; execute table entry
002623' 254 00 0 00 002652'                      JRST   TX54                    ; end of test execute table
002624' 254 00 0 00 002622'                      JRST   TA54                    ; keep looping after call
002625' 474 15 0 00 000000                       SETO   ERFLG,                  ; error occurred

                                        ; Get miscellaneous data

002626' 201 01 0 00 000300                      MOVEI   1,300                   ; get start address
002627' 242 01 0 00 000001                      LSH     1,1                     ; position correctly
002630' 260 17 0 00 002277*                     GO      LDRAR                   ; load the register
002631' 201 01 0 00 220010                      MOVEI   1,TSTEBF!MPRUN!SINCYC   ; get start data
002632' 260 17 0 00 002305*                     GO      LDCSR                   ; start up the port
002633' 260 17 0 00 002260*                     GO      RDEBUF                  ; read EBUF data
002634' 202 01 0 00 005730'                     MOVEM   1,SAVSEG                ; save segment number
002635' 202 01 0 00 002620*                     MOVEM   1,TSTSUB                ; set up subtest number
002636' 350 00 0 00 002635*                     AOS     TSTSUB                  ;   ...

002637' 201 01 0 00 000200                      MOVEI   1,200                   ; get start address
002640' 242 01 0 00 000001                      LSH     1,1                     ; position correctly
002641' 260 17 0 00 002630*                     GO      LDRAR                   ; load the register
002642' 201 01 0 00 220010                      MOVEI   1,TSTEBF!MPRUN!SINCYC   ; get start data
002643' 260 17 0 00 002632*                     GO      LDCSR                   ; start up the port
002644' 260 17 0 00 002633*                     GO      RDEBUF                  ; read EBUF data
002645' 202 01 0 00 005723'                     MOVEM   1,SAVDAT                ; save data read

                                        ; Handle error printouts and scope looping

002646' 027 00 0 00 003000'                     SCOPER  MA54                    ; print error message
002647' 254 00 0 00 002616'                     JRST    TL54                    ; loop on error
002650' 254 00 0 00 002652'                     JRST    TX54                    ; altmode exit
002651' 322 15 0 00 002622'                     JUMPE   ERFLG,TA54              ; do next test execute table entry

                                        ; End of test

002652' 263 17 0 00 000000              TX54:   RTN                             ; return

                                        ; Test Execute Table, as:  (CMD,parameters)

002653' 140000  002657'                 TS54:   TTABLE  (TCALL,TS54DA)          ; set up data transfer
002654' 040000  001000                          TTABLE  (TSTART,1000)           ; start up port
002655' 240002  000000                          TTABLE  (TCHECK,2)              ; check if completed
002656' 000000  000000                          TTABLE  (TLAST)                 ; end of table

```

```


                                       ; Set up data transfer

 002657' 400 01 0 00 000000            TS54DA: SETZ    1,                           ; clear CSR data
 002660' 260 17 0 00 002643*                   GO      LDCSR                        ; write CSR register
 002661' 201 01 0 00 005000                    MOVEI   1,5000                       ; get start address
 002662' 242 01 0 00 000001                    LSH     1,1                          ; position correctly
 002663' 260 17 0 00 002641*                   GO      LDRAR                        ; load the register
 002664' 201 01 0 00 000010                    MOVEI   1,MPRUN                      ; get start data
 002665' 260 17 0 00 002660*                   GO      LDCSR                        ; start up the port
 002666' 201 00 0 00 000001                    MOVEI   1                            ; wait a bit (1 msec)
 002667' 260 17 0 00 002303*                   GO      ODELAY
 002670' 400 01 0 00 000000                    SETZ    1,                           ; clear CSR data
 002671' 260 17 0 00 002665*                   GO      LDCSR                        ; stop the port
 002672' 200 01 0 00 005103'                   MOVE    1,[CBUFSZ,,CBUF]             ; get pointer to control buffer
 002673' 260 17 0 00 002307*                   GO      CHINIT                       ; initialize software
 002674' 551 01 0 00 002310*                   HRRZI   1,BUFF                       ; buffer address
 002675' 201 02 0 00 000074                    MOVEI   2,^D60                       ; word count
 002676' 400 03 0 00 000000                    SETZ    3,                           ; clear zero fill specifier
 002677' 260 17 0 00 002313*                   GO      GENCCW                       ; generate a CCW list
 002700' 200 01 0 00 005577'                   MOVE    1,[TS54BU,,BUFF]             ; copy the data for this test
 002701' 251 01 0 00 000000#                   BLT     1,BUFF+^D59                  ;   into the data buffer
 002702' 476 00 0 00 005730'                   SETOM   SAVSEG#                      ; initialize test segment
 002703' 263 17 0 00 000000                    RTN                                  ; return

                                       ; Data

 002704' 400001  000000                TS54BU: 400001,,0
 002705' 000000  000000                        0
 002706' 000000  000020                        20
 002707' 000000  000020                        20
 002710' 000001  000000                        1,,0
 002711' 000001  000000                        1,,0
 002712' 400003  000020                        400003,,20
 002713' 000001  000020                        1,,20
 002714' 400001  000000                        400001,,0
 002715' 400001  000020                        400001,,20
 002716' 400003  000000                        400003,,0
 002717' 000001  000020                        1,,20

 002720' 400001  000001                        400001,,1
 002721' 000000  000001                        1
 002722' 000000  000021                        21
 002723' 000000  000021                        21
 002724' 000001  000001                        1,,1
 002725' 000001  000001                        1,,1
 002726' 400003  000021                        400003,,21
 002727' 000001  000021                        1,,21
 002730' 400001  000001                        400001,,1
 002731' 400001  000021                        400001,,21
 002732' 400003  000001                        400003,,1
 002733' 000001  000021                        1,,21
```

```

002734' 400001  000002                  400001,,2
002735' 000000  000002                  2
002736' 000000  000022                  22
002737' 000000  000022                  22
002740' 000001  000002                  1,,2
002741' 000001  000002                  1,,2
002742' 400003  000022                  400003,,22
002743' 000001  000022                  1,,22
002744' 400001  000002                  400001,,2
002745' 400001  000022                  400001,,22
002746' 400003  000002                  400003,,2
002747' 000001  000022                  1,,22

002750' 400001  000004                  400001,,4
002751' 000000  000004                  4
002752' 000000  000024                  24
002753' 000000  000024                  24
002754' 000001  000004                  1,,4
002755' 000001  000004                  1,,4
002756' 400003  000024                  400003,,24
002757' 000001  000024                  1,,24
002760' 400001  000004                  400001,,4
002761' 400001  000024                  400001,,24
002762' 400003  000004                  400003,,4
002763' 000001  000024                  1,,24

002764' 400001  000010                  400001,,10
002765' 000000  000010                  10
002766' 000000  000030                  30
002767' 000000  000030                  30
002770' 000001  000010                  1,,10
002771' 000001  000010                  1,,10
002772' 400003  000030                  400003,,30
002773' 000001  000030                  1,,30
002774' 400001  000010                  400001,,10
002775' 400001  000030                  400001,,30
002776' 400003  000010                  400003,,10
002777' 000001  000030                  1,,30
```

```

                                         ; Error messages

 003000' 140000  005524'         MA54:   MSG!TXNOT![ASCIZ /Parity Predictor produced incorrect result:/]
 003001' 240000  003003'                 CALL!TXNOT!MA54PN                  ; print result
 003002' 000000000000#                   LAST!CALL!TXALL!TTPNT              ; print test data

 003003' 200 01 0 00 005730'     MA54PN: MOVE    1,SAVSEG                   ; get test segment
 003004' 037 00 0 00 005535'             TMSGC   <  Data in CBIN:  >
 003005' 200 00 0 01 002704'             MOVE    TS54BU(1)                  ; get data word
 003006' 037 13 0 00 000000              PNTHW                              ; print it
 003007' 200 01 0 01 002704'             MOVE    1,TS54BU(1)                ; get data word
 003010' 607 01 0 00 400000              TLNN    1,400000                   ; should be cleared?
 003011' 037 00 0 00 005541'             TMSGC   <  'CCMVPARCHK' should not be asserted, but is>
 003012' 603 01 0 00 400000              TLNE    1,400000                   ; should be set?
 003013' 037 00 0 00 005553'             TMSGC   <  'CCMVPARCHK' should be asserted, but isn't>
 003014' 263 17 0 00 000000              RTN                                ; return

                                 ; Microcode

                                 ;       Read 1st word

 003015' 100000  000030          T54M:   MWORD   <ADDR=1000,JZ,SELM,MGC=30>                      ; 1000
 003016' 000000  002000
 003017' 000000  100000                  MWORD   <ADDR=0,LDCT,J=10>                              ; 0
 003020' 000000  000300
 003021' 000100  010000                  MWORD   <RPCT,J=1>                                      ; 1
 003022' 000000  000220
 003023' 000200  000000                  MWORD   <CONT,SOB,AND,B=5,D=2>                          ; 2
 003024' 342002  400340
 003025' 000301  000000                  MWORD   <CJS,J=100,D=1>                                 ; 3
 003026' 001000  000020
 003027' 000400  060000                  MWORD   <JMAP,J=6>                                      ; 4
 003030' 000000  000040


                                 ;       Read 2nd through nth words

 003031' 000501  030000                  MWORD   <CJS,J=103,D=1>                                 ; 5
 003032' 001000  000020
 003033' 000600  110000                  MWORD   <CJP,J=11,SOA,OR,A=3,D=1,CENA,CCMB>             ; 6
 003034' 031430  120060
 003035' 000752  000000                  MWORD   <CJP,J=5200,D=1,CENA,CCMP>                      ; 7
 003036' 001400  130060
 003037' 001000  130000                  MWORD   <JMAP,J=13,D=1>                                 ; 10
 003040' 001000  000040
 003041' 001100  130000                  MWORD   <CJP,J=13,D=1,CENA,CCMP>                        ; 11
 003042' 001400  130060
 003043' 001252  000000                  MWORD   <JMAP,J=5200,D=1>                               ; 12
 003044' 001000  000040
 003045' 001300  150000                  MWORD   <CJP,J=15,SOA,PLUS,A=5,B=5,D=2,CRY,CENA,CCLW>   ; 13
 003046' 402452  570460
 003047' 001400  050000                  MWORD   <JMAP,J=5>                                      ; 14
 003050' 000000  000040

                                 ;       All done -  set STOP and STORE
```

```

003051' 001551  000140                  MWORD   <JMAP,J=5100,SELC,MGC=140>                      ; 15
003052' 000000  004040

                                ;       Routine to read a word from CBUS, through formatter into R3
                                ;       (If this routine is started at 101, then the next word from
                                ;       the CBUS is read).

003053' 010000  000200                  MWORD   <ADDR=100,CONT,D=1,SELC,MGC=200>                ; 100
003054' 001000  004340
003055' 010100  100000                  MWORD   <LDCT,J=10,D=1>                                 ; 101
003056' 001000  000300
003057' 010201  020000                  MWORD   <RPCT,J=102,D=1>                                ; 102
003060' 001000  000220
003061' 010301  050000                  MWORD   <CJP,J=105,D=1,CENA,CCCA>                       ; 103
003062' 001400  000060
003063' 010401  030000                  MWORD   <JMAP,J=103,D=1>                                ; 104
003064' 001000  000040
003065' 010500  000014                  MWORD   <CONT,D=1,SELC,MGC=14>                          ; 105
003066' 001000  004340
003067' 010600  000200                  MWORD   <CONT,SDO,OR,B=3,D=2,SELM,MGC=200>              ; 106
003070' 732001  402340
003071' 010700  000000                  MWORD   <CRTN,D=1>                                      ; 107
003072' 001000  000240

                                ;       Routine to read R3 and place in EBUF

003073' 020002  002004                  MWORD   <ADDR=200,JMAP,J=200,SOA,A=3,OR,OENA,D=1,SELE,MGC=4>
003074' 431030  005040

                                ;       Routine to read R5 and place in EBUF

003075' 030003  002004                  MWORD   <ADDR=300,JMAP,J=300,SOA,A=5,OR,OENA,D=1,SELE,MGC=4>
003076' 431050  005040

                                ;       Initialization step

003077' 500050  010100                  MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>            ; 5000
003100' 000000  004040
003101' 500150  010020                  MWORD   <JMAP,J=5001,SELM,MGC=20>                       ; 5001
003102' 000000  002040

                                ; Subroutine - Passed - Write 1 to bit 12 of CSR register

003103' 510000  000000                  MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>        ; 5100
003104' 742001  000340
003105' 510100  000000                  MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5101
003106' 302001  000740
003107' 510200  260000                  MWORD   <LDCT,J=26>                             ; 5102
003110' 000000  000300
003111' 510351  030000                  MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>      ; 5103
003112' 102021  000220
003113' 510400  002004                  MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5104
003114' 431020  005340
003115' 510551  070040                  MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>  ; 5105
```

```
003116' 001400  015060
003117' 510651  050000          MWORD   <JMAP,J=5105,D=1>                       ; 5106
003120' 001000  000040
003121' 510700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5107
003122' 431010  005340
003123' 511051  100000          MWORD   <JMAP,J=5110,D=1>                       ; 5110
003124' 001000  000040

                        ; Subroutine - Failed - Write 1 to bit 13 of CSR register

003125' 520000  000000          MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>        ; 5200
003126' 742001  000340
003127' 520100  000000          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5201
003130' 302001  000740
003131' 520200  250000          MWORD   <LDCT,J=25>                             ; 5202
003132' 000000  000300
003133' 520352  030000          MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>      ; 5203
003134' 102021  000220
003135' 520400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5204
003136' 431020  005340
003137' 520552  070040          MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>  ; 5205
003140' 001400  015060
003141' 520652  050000          MWORD   <JMAP,J=5205,D=1>                       ; 5206
003142' 001000  000040
003143' 520700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5207
003144' 431010  005340
003145' 521052  100000          MWORD   <JMAP,J=5210,D=1>                       ; 5210
003146' 001000  000040
003147' 777777  777777          -1
```

```
                                ;#*********************************************************************
                                ;* Test 55 - Parity Predictor - Fmtr to PLOUT
                                ;
                                ; Description:   Verify that the parity predictor works  correctly
                                ;                when moving data from formatter to PLOUT.
                                ;
                                ; Procedure:     KL> Port Clear
                                ;                KL> Set up a CCW list to transfer 8 words
                                ;                KL> Start up port
                                ;                UC> Read words from CBUS into CBIN to formatter
                                ;                    then to local storage locations 0-7.
                                ;                UC> Initialize parity predictor with 'MPCLRPARCHK'
                                ;
                                ;                UC> Load formatter with word from local storage
                                ;                UC> Assert 'MPFMTRTOPLOUT'
                                ;                UC> Test leftmost bit of data word - this should
                                ;                    be the setting of 'CCMVRPARCHK'.
                                ;                UC> if correct - set bit 12 (DISABLE Complete)
                                ;                UC> if not correct - set bit 13 (ENABLE Complete)
                                ;                Read CSR and determine test disposition
                                ;
                                ;                Repeat the above steps for each of the 8 data
                                ;                words.
                                ;
                                ; Failure:       ---
                                ;#*********************************************************************

                                ; Test data

 003150' 254 00 0 00 003160'    TSTC55: JRST    TG55                            ; go start test
 003151' 100604  000055                 CBUS!NDMP!NDCB!ZCBUS!55                 ; test mask
 003152' 003300' 005600'                T55M,,[ASCIZ ^Parity Predictor - Fmtr to PLOUT^]
 003153' 005517' 005520'                [EXP MLAST!C11],,[EXP C12,C14,MLAST!C21]
 003154' 000000  003471'                TSTC56                                  ; failure test table
 003155' 000000  004013'                TSTC57                                  ;   ...
 003156' 000000  004352'                TSTC60
 003157' 777777  777777                 -1

                                ; Start test

 003160' 201 00 0 00 000000'    TG55:   MOVEI   ZA                              ; get address of module start
 003161' 260 17 0 00 002612*            GO      TRACE                           ; handle trace output
 003162' 201 01 0 00 003300'            MOVEI   1,T55M                          ; set up microcode address
 003163' 260 17 0 00 002614*            GO      TLOAD                           ; load/verify it
 003164' 263 17 0 00 000000             RTN                                     ; failed - exit test

                                ; Initialization

 003165' 400 15 0 00 000000     TL55:   SETZ    ERFLG,                          ; clear error flag
 003166' 260 17 0 00 002617*            GO      IPACLR                          ; clear port
 003167' 402 00 0 00 002636*            SETZM   TSTSUB                          ; initialize subtest number
 003170' 201 06 0 00 003202'            MOVEI   6,TS55                          ; get test table address
```

```

                                  ; Loop on test execute table entries

003171' 260 17 0 00 002622*       TA55:   GO      TEXEC                   ; execute table entry
003172' 254 00 0 00 003201'               JRST    TX55                    ; end of test execute table
003173' 254 00 0 00 003171'               JRST    TA55                    ; keep looping after call
003174' 474 15 0 00 000000                SETO    ERFLG,                  ; error occurred

                                  ; Handle error printouts and scope looping

003175' 027 00 0 00 003263'               SCOPER  MA55                    ; print error message
003176' 254 00 0 00 003165'               JRST    TL55                    ; loop on error
003177' 254 00 0 00 003201'               JRST    TX55                    ; altmode exit
003200' 322 15 0 00 003171'               JUMPE   ERFLG,TA55              ; do next test execute table entry

                                  ; End of test

003201' 263 17 0 00 000000        TX55:   RTN                             ; return

                                  ; Test Execute Table, as:  (CMD,parameters)

003202' 140000  003212'           TS55:   TTABLE  (TCALL,TS55DA)          ; set up data transfer
003203' 140000  003243'                   TTABLE  (TCALL,TS55IN)          ; initialize counter
003204' 200000  003245'           TS55A:  TTABLE  (TCALLC,TS55NE)         ; set up next test segment
003205' 400000  000010                    TTABLE  (TSSTAR,10)             ; start up port (read a word)
003206' 240002  000000                    TTABLE  (TCHECK,2)              ; check if completed
003207' 340000  000000                    TTABLE  (TEXIT)                 ; exit if error yet
003210' 300000  003204'                   TTABLE  (TJRST,TS55A)           ; loop till done
003211' 000000  000000                    TTABLE  (TLAST)                 ; end of table

                                  ; Set up data transfer

003212' 201 01 0 00 000200        TS55DA: MOVEI   1,200                   ; get start address
003213' 242 01 0 00 000001                LSH     1,1                     ; position correctly
003214' 260 17 0 00 002663*               GO      LDRAR                   ; load the register
003215' 201 01 0 00 000010                MOVEI   1,MPRUN                 ; get start data
003216' 260 17 0 00 002671*               GO      LDCSR                   ; start up the port
003217' 200 01 0 00 005103'               MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
003220' 260 17 0 00 002673*               GO      CHINIT                  ; initialize software
003221' 551 01 0 00 002674*               HRRZI   1,BUFF                  ; buffer address
003222' 201 02 0 00 000010                MOVEI   2,^D8                   ; word count
003223' 400 03 0 00 000000                SETZ    3,                      ; clear zero fill specifier
003224' 260 17 0 00 002677*               GO      GENCCW                  ; generate a CCW list
003225' 200 01 0 00 005607'               MOVE    1,[TS55BU,,BUFF]        ; copy the data for this test
003226' 251 01 0 00 000000#               BLT     1,BUFF+7                ;   into the data buffer
003227' 400 01 0 00 000000                SETZ    1,                      ; clear CSR data
003230' 260 17 0 00 003216*               GO      LDCSR                   ; write CSR register
003231' 201 01 0 00 005000                MOVEI   1,5000                  ; get start address
003232' 242 01 0 00 000001                LSH     1,1                     ; position correctly
003233' 260 17 0 00 003214*               GO      LDRAR                   ; load the register
```

```
003234' 201 01 0 00 000010            MOVEI   1,MPRUN                 ; get start data
003235' 260 17 0 00 003230*           GO      LDCSR                   ; start up the port
003236' 201 00 0 00 000001            MOVEI   1                       ; ensure the port has enough
003237' 260 17 0 00 002667*           GO      ODELAY                  ;   time to initialize
003240' 400 01 0 00 000000            SETZ    1,                      ; clear CSR data
003241' 260 17 0 00 003235*           GO      LDCSR                   ; stop the port
003242' 263 17 0 00 000000            RTN                             ; return

                              ; Initialize test segment

003243' 476 00 0 00 005730'   TS55IN: SETOM   SAVSEG#                 ; initialize test segment
003244' 263 17 0 00 000000            RTN                             ; return

                              ; Point to next test segment

003245' 350 01 0 00 005730'   TS55NE: AOS     1,SAVSEG                ; point to next test segment
003246' 307 01 0 00 000007            CAIG    1,^D7                   ; done yet?
003247' 350 00 0 17 000000            AOS     (P)                     ; no - set up proper return
003250' 271 01 0 00 000010            ADDI    1,10                    ; point to starting address
003251' 542 01 0 00 003205'           HRRM    1,TS55A+1               ; set up starting address
003252' 263 17 0 00 000000            RTN                             ; return

                              ; Data

003253' 400000  000000        TS55BU: 400000,,0
003254' 000000  000003                0,,3
003255' 400000  000005                400000,,5
003256' 000000  000011                0,,11
003257' 400000  000001                400000,,1
003260' 000000  000002                0,,2
003261' 400000  000004                400000,,4
003262' 000000  000010                0,,10

                              ; Error messages

003263' 140000  005524'       MA55:   MSG!TXNOT![ASCIZ /Parity Predictor produced incorrect result:/]
003264' 240000  003266'               CALL!TXNOT!MA55PN               ; print result
003265' 000000000000#                 LAST!CALL!TXALL!TTPNT           ; print test data

003266' 200 01 0 00 005730'   MA55PN: MOVE    1,SAVSEG                ; get test segment
003267' 037 00 0 00 005610'           TMSGC   <  Data in FMTR:  >
003270' 200 00 0 01 003253'           MOVE    TS55BU(1)               ; get data word
003271' 037 13 0 00 000000            PNTHW                           ; print it
003272' 200 01 0 01 003253'           MOVE    1,TS55BU(1)             ; get data word
003273' 607 01 0 00 400000            TLNN    1,400000                ; should be cleared?
003274' 037 00 0 00 005541'           TMSGC   <  'CCMVPARCHK' should not be asserted, but is>
003275' 603 01 0 00 400000            TLNE    1,400000                ; should be set?
003276' 037 00 0 00 005553'           TMSGC   <  'CCMVPARCHK' should be asserted, but isn't>
003277' 263 17 0 00 000000            RTN                             ; return
```

```

                                  ; Microcode

                                  ;       Read local storage location

003300' 001000  270000            T55M:   MWORD   <ADDR=10,JMAP,J=27,SD0,OR,B=3,D=2,RDLM,MGC=0>
003301' 732001  620040
003302' 001100  300001                    MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=1>   ; 11
003303' 732001  620040
003304' 001200  300002                    MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=2>   ; 12
003305' 732001  620040
003306' 001300  300003                    MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=3>   ; 13
003307' 732001  620040
003310' 001400  300004                    MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=4>   ; 14
003311' 732001  620040
003312' 001500  300005                    MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=5>   ; 15
003313' 732001  620040
003314' 001600  300006                    MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=6>   ; 16
003315' 732001  620040
003316' 001700  300007                    MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=7>   ; 17
003317' 732001  620040

                                  ;       Do a test segment

003320' 002700  000030                    MWORD   <ADDR=27,CONT,D=1,SELM,MGC=30>          ; 27
003321' 001000  002340
003322' 003000  002040                    MWORD   <CONT,SOA,A=3,D=1,OENA,SELF,MGC=40>     ; 30
003323' 401030  003340
003324' 003100  000010                    MWORD   <CONT,D=1,SELF,MGC=10>                  ; 31
003325' 001000  003340
003326' 003200  350000                    MWORD   <CJP,J=35,SOA,A=3,D=1,CENA,CCMB>        ; 32
003327' 401430  120060
003330' 003352  000000                    MWORD   <CJP,J=5200,D=1,CENA,CCMP>              ; 33
003331' 001400  130060
003332' 003451  000000                    MWORD   <JMAP,J=5100,D=1>                       ; 34
003333' 001000  000040
003334' 003551  000000                    MWORD   <CJP,J=5100,D=1,CENA,CCMP>              ; 35
003335' 001400  130060
003336' 003652  000000                    MWORD   <JMAP,J=5200,D=1>                       ; 36
003337' 001000  000040

                                  ;       Read a word from CBUS into CBIN

003340' 010001  010200                    MWORD   <ADDR=100,JMAP,J=101,D=1,SELC,MGC=200>  ; 100
003341' 001000  004040
003342' 010101  030000                    MWORD   <CJP,J=103,D=1,CENA,CCCA>               ; 101
003343' 001400  000060
003344' 010201  010000                    MWORD   <JMAP,J=101,D=1>                        ; 102
003345' 001000  000040
003346' 010300  000004                    MWORD   <CONT,D=1,SELC,MGC=4>                   ; 103
003347' 001000  004340
003350' 010400  000200                    MWORD   <CRTN,SD0,OR,B=3,D=2,SELM,MGC=200>      ; 104
003351' 732001  402240

                                  ;       Ensure CBUS is stopped
```

```

003352' 020000  000120          MWORD   <ADDR=200,JZ,SELC,MGC=120>              ; 200
003353' 000000  004000


                        ;       Initialization steps

003354' 500050  010100          MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>    ; 5000
003355' 000000  004040
003356' 500100  000020          MWORD   <CONT,SELM,MGC=20>                      ; 5001
003357' 000000  002340
003360' 500201  000000          MWORD   <CJS,J=100,D=1>                         ; 5002
003361' 001000  000020
003362' 500300  002000          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=0>      ; 5003
003363' 401030  230340
003364' 500401  010000          MWORD   <CJS,J=101,D=1>                         ; 5004
003365' 001000  000020
003366' 500500  002001          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=1>      ; 5005
003367' 401030  230340
003370' 500601  010000          MWORD   <CJS,J=101,D=1>                         ; 5006
003371' 001000  000020
003372' 500700  002002          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=2>      ; 5007
003373' 401030  230340
003374' 501001  010000          MWORD   <CJS,J=101,D=1>                         ; 5010
003375' 001000  000020
003376' 501100  002003          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=3>      ; 5011
003377' 401030  230340
003400' 501201  010000          MWORD   <CJS,J=101,D=1>                         ; 5012
003401' 001000  000020
003402' 501300  002004          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=4>      ; 5013
003403' 401030  230340
003404' 501401  010000          MWORD   <CJS,J=101,D=1>                         ; 5014
003405' 001000  000020
003406' 501500  002005          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=5>      ; 5015
003407' 401030  230340
003410' 501601  010000          MWORD   <CJS,J=101,D=1>                         ; 5016
003411' 001000  000020
003412' 501700  002006          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=6>      ; 5017
003413' 401030  230340
003414' 502001  010000          MWORD   <CJS,J=101,D=1>                         ; 5020
003415' 001000  000020
003416' 502100  002007          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=7>      ; 5021
003417' 401030  230340
003420' 502200  000000          MWORD   <JZ>                                    ; 5022
003421' 000000  000000
003422' 000000  000000          MWORD   <ADDR=0,JZ>                             ; 5023
003423' 000000  000000

                        ; Subroutine - Passed - Write 1 to bit 12 of CSR register

003424' 510000  000000          MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>        ; 5100
003425' 742001  000340
003426' 510100  000000          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5101
003427' 302001  000740
003430' 510200  260000          MWORD   <LDCT,J=26>                             ; 5102
003431' 000000  000300
```

```
2921  003432' 510351  030000            MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>        ; 5103
2922  003433' 102021  000220
2923  003434' 510400  002004            MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>     ; 5104
2924  003435' 431020  005340
2925  003436' 510551  070040            MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>    ; 5105
2926  003437' 001400  015060
2927  003440' 510651  050000            MWORD   <JMAP,J=5105,D=1>                         ; 5106
2928  003441' 001000  000040
2929  003442' 510700  000200            MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>        ; 5107
2930  003443' 431010  005340
2931  003444' 511051  100000            MWORD   <JMAP,J=5110,D=1>                         ; 5110
2932  003445' 001000  000040
2933
2934                              ; Subroutine - Failed - Write 1 to bit 13 of CSR register
2935
2936  003446' 520000  000000            MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>          ; 5200
2937  003447' 742001  000340
2938  003450' 520100  000000            MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>               ; 5201
2939  003451' 302001  000740
2940  003452' 520200  250000            MWORD   <LDCT,J=25>                               ; 5202
2941  003453' 000000  000300
2942  003454' 520352  030000            MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>        ; 5203
2943  003455' 102021  000220
2944  003456' 520400  002004            MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>     ; 5204
2945  003457' 431020  005340
2946  003460' 520552  070040            MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>    ; 5205
2947  003461' 001400  015060
2948  003462' 520652  050000            MWORD   <JMAP,J=5205,D=1>                         ; 5206
2949  003463' 001000  000040
2950  003464' 520700  000200            MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>        ; 5207
2951  003465' 431010  005340
2952  003466' 521052  100000            MWORD   <JMAP,J=5210,D=1>                         ; 5210
2953  003467' 001000  000040
2954  003470' 777777  777777            -1
```

```

                                        ;#*******************************************************************
                                        ;* Test 56 - Parity Predictor - PLIN to Fmtr/MPCOREDUMP=0
                                        ;
                                        ; Description:  Verify that the parity predictor works  correctly
                                        ;               when  moving  data  from  PLIN  to formatter with
                                        ;               'MPCOREDUMP' not asserted. This verifies one half
                                        ;               of the parity predictor
                                        ;
                                        ; Procedure:    KL> Port Clear
                                        ;               KL> Set up a CCW list to transfer 8 words
                                        ;               KL> Start up port
                                        ;               UC> Read words from CBUS into CBIN to formatter
                                        ;                   then to local storage locations 0-7.
                                        ;               UC> Initialize parity predictor with 'MPCLRPARCHK'
                                        ;
                                        ;               UC> Load PLOUT with word from local storage
                                        ;               UC> Load PLIN with word from PLOUT
                                        ;               UC> Assert 'MPPLINTOFMTR'
                                        ;               UC> Test leftmost bit of data word - this should
                                        ;                   be the setting of 'CCMVRPARCHK'.
                                        ;               UC> if correct - set bit 12 (DISABLE Complete)
                                        ;               UC> if not correct - set bit 13 (ENABLE Complete)
                                        ;               Read CSR and determine test disposition
                                        ;
                                        ;               Repeat the above steps for each of the 8 data
                                        ;               words.
                                        ;
                                        ; Failure:      ---
                                        ;#*******************************************************************

                                        ; Test data

003471' 254 00 0 00 003500'             TSTC56: JRST    TG56                    ; go start test
003472' 100604  000056                          CBUS!NDMP!NDCB!ZCBUS!56         ; test mask
003473' 003620' 005614'                         T56M,,[ASCIZ ^Parity Predictor - PLIN to Fmtr/MPCOREDUMP=0^]
003474' 005517' 005625'                         [EXP MLAST!C11],,[EXP C8,C12,C14,MLAST!C21]
003475' 000000  004013'                         TSTC57                          ; failure test table
003476' 000000  004352'                         TSTC60                          ;   ...
003477' 777777  777777                          -1

                                        ; Start test

003500' 201 00 0 00 000000'             TG56:   MOVEI   ZA                      ; get address of module start
003501' 260 17 0 00 003161*                     GO      TRACE                   ; handle trace output
003502' 201 01 0 00 003620'                     MOVEI   1,T56M                  ; set up microcode address
003503' 260 17 0 00 003163*                     GO      TLOAD                   ; load/verify it
003504' 263 17 0 00 000000                      RTN                             ; failed - exit test
```

```

                                           ; Initialization

003505' 400 15 0 00 000000                 TL56:   SETZ    ERFLG,                  ; clear error flag
003506' 260 17 0 00 003166*                        GO      IPACLR                  ; clear port
003507' 402 00 0 00 003167*                        SETZM   TSTSUB                  ; initialize subtest number
003510' 201 06 0 00 003522'                        MOVEI   6,TS56                  ; get test table address

                                           ; Loop on test execute table entries

003511' 260 17 0 00 003171*                TA56:   GO      TEXEC                   ; execute table entry
003512' 254 00 0 00 003521'                         JRST   TX56                    ; end of test execute table
003513' 254 00 0 00 003511'                         JRST   TA56                    ; keep looping after call
003514' 474 15 0 00 000000                          SETO   ERFLG,                  ; error occurred

                                           ; Handle error printouts and scope looping

003515' 027 00 0 00 003603'                        SCOPER  MA56                    ; print error message
003516' 254 00 0 00 003505'                        JRST    TL56                    ; loop on error
003517' 254 00 0 00 003521'                        JRST    TX56                    ; altmode exit
003520' 322 15 0 00 003511'                        JUMPE   ERFLG,TA56              ; do next test execute table entry

                                           ; End of test

003521' 263 17 0 00 000000                 TX56:   RTN                             ; return

                                           ; Test Execute Table, as:  (CMD,parameters)

003522' 140000  003532'                    TS56:   TTABLE  (TCALL,TS56DA)          ; set up data transfer
003523' 140000  003563'                            TTABLE  (TCALL,TS56IN)          ; initialize counter
003524' 200000  003565'                    TS56A:  TTABLE  (TCALLC,TS56NE)         ; set up next test segment
003525' 400000  000010                             TTABLE  (TSSTAR,10)             ; start up port (read a word)
003526' 240002  000000                             TTABLE  (TCHECK,2)              ; check if completed
003527' 340000  000000                             TTABLE  (TEXIT)                 ; exit if error yet
003530' 300000  003524'                            TTABLE  (TJRST,TS56A)           ; loop till done
003531' 000000  000000                             TTABLE  (TLAST)                 ; end of table

                                           ; Set up data transfer

003532' 201 01 0 00 000200                 TS56DA: MOVEI   1,200                   ; get start address
003533' 242 01 0 00 000001                         LSH     1,1                     ; position correctly
003534' 260 17 0 00 003233*                        GO      LDRAR                   ; load the register
003535' 201 01 0 00 000010                         MOVEI   1,MPRUN                 ; get start data
003536' 260 17 0 00 003241*                        GO      LDCSR                   ; start up the port
003537' 200 01 0 00 005103'                        MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
003540' 260 17 0 00 003220*                        GO      CHINIT                  ; initialize software
003541' 551 01 0 00 003221*                        HRRZI   1,BUFF                  ; buffer address
003542' 201 02 0 00 000010                         MOVEI   2,^D8                   ; word count
003543' 400 03 0 00 000000                         SETZ    3,                      ; clear zero fill specifier
003544' 260 17 0 00 003224*                        GO      GENCCW                  ; generate a CCW list
```

```
003545' 200 01 0 00 005631'                MOVE    1,[TS56BU,,BUFF]        ; copy the data for this test
003546' 251 01 0 00 000000#                BLT     1,BUFF+7                ;   into the data buffer
003547' 400 01 0 00 000000                 SETZ    1,                      ; clear CSR data
003550' 260 17 0 00 003536*                GO      LDCSR                   ; write CSR register
003551' 201 01 0 00 005000                 MOVEI   1,5000                  ; get start address
003552' 242 01 0 00 000001                 LSH     1,1                     ; position correctly
003553' 260 17 0 00 003534*                GO      LDRAR                   ; load the register
003554' 201 01 0 00 000010                 MOVEI   1,MPRUN                 ; get start data
003555' 260 17 0 00 003550*                GO      LDCSR                   ; start up the port
003556' 201 00 0 00 000001                 MOVEI   1                       ; ensure the port has enough
003557' 260 17 0 00 003237*                GO      ODELAY                  ;   time to initialize
003560' 400 01 0 00 000000                 SETZ    1,                      ; clear CSR data
003561' 260 17 0 00 003555*                GO      LDCSR                   ; stop the port
003562' 263 17 0 00 000000                 RTN                             ; return

                                   ; Initialize test segment

003563' 476 00 0 00 005730'        TS56IN: SETOM   SAVSEG#                 ; initialize test segment
003564' 263 17 0 00 000000                 RTN                             ; return

                                   ; Point to next test segment

003565' 350 01 0 00 005730'        TS56NE: AOS     1,SAVSEG                ; point to next test segment
003566' 307 01 0 00 000007                 CAIG    1,^D7                   ; done yet?
003567' 350 00 0 17 000000                 AOS     (P)                     ; no - set up proper return
003570' 271 01 0 00 000010                 ADDI    1,10                    ; point to starting address
003571' 542 01 0 00 003525'                HRRM    1,TS56A+1               ; set up starting address
003572' 263 17 0 00 000000                 RTN                             ; return

                                   ; Data

003573' 400000  000000             TS56BU: 400000,,0
003574' 000000  000000                     0,,0
003575' 400000  000000                     400000,,0
003576' 000000  000000                     0,,0
003577' 000000  000001                     0,,1
003600' 400000  000001                     400000,,1
003601' 000000  000001                     0,,1
003602' 400000  000001                     400000,,1
```

```
                                              ; Error messages

003603' 140000  005524'                       MA56:   MSG!TXNOT![ASCIZ /Parity Predictor produced incorrect result:/]
003604' 240000  003606'                               CALL!TXNOT!MA56PN              ; print result
003605' 000000000000#                                 LAST!CALL!TXALL!TTPNT          ; print test data

003606' 200 01 0 00 005730'                   MA56PN: MOVE    1,SAVSEG               ; get test segment
003607' 037 00 0 00 005632'                           TMSGC   <  Data in PLIN:  >
003610' 550 00 0 01 003573'                           HRRZ    TS56BU(1)              ; get data word
003611' 037 03 0 00 000000                            PNT3                           ; print it
003612' 200 01 0 01 003573'                           MOVE    1,TS56BU(1)            ; get data word
003613' 607 01 0 00 400000                            TLNN    1,400000               ; should be cleared?
003614' 037 00 0 00 005541'                           TMSGC   <  'CCMVPARCHK' should not be asserted, but is>
003615' 603 01 0 00 400000                            TLNE    1,400000               ; should be set?
003616' 037 00 0 00 005553'                           TMSGC   <  'CCMVPARCHK' should be asserted, but isn't>
003617' 263 17 0 00 000000                            RTN                            ; return

                                              ; Microcode

                                              ;       Read local storage location

003620' 001000  270000                        T56M:   MWORD   <ADDR=10,JMAP,J=27,SD0,OR,B=3,D=2,RDLM,MGC=0>
003621' 732001  620040
003622' 001100  300001                                MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=1>   ; 11
003623' 732001  620040
003624' 001200  300002                                MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=2>   ; 12
003625' 732001  620040
003626' 001300  300003                                MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=3>   ; 13
003627' 732001  620040
003630' 001400  300004                                MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=4>   ; 14
003631' 732001  620040
003632' 001500  300005                                MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=5>   ; 15
003633' 732001  620040
003634' 001600  300006                                MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=6>   ; 16
003635' 732001  620040
003636' 001700  300007                                MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=7>   ; 17
003637' 732001  620040

                                              ;       Do a test segment

003640' 002700  000030                                MWORD   <ADDR=27,CONT,D=1,SELM,MGC=30>          ; 27
003641' 001000  002340
003642' 003000  002040                                MWORD   <CONT,SOA,A=3,D=1,OENA,SELM,MGC=40>     ; 30
003643' 401030  002340
003644' 003100  000006                                MWORD   <CONT,SD0,D=1,SELP,MGC=6>               ; 31
003645' 701000  001340
003646' 003200  000020                                MWORD   <CONT,D=1,SELF,MGC=20>                  ; 32
003647' 001000  003340
003650' 003300  350000                                MWORD   <CJP,J=35,SOA,A=3,D=1,CENA,CCMB>        ; 32
003651' 401430  120060
003652' 003452  000000                                MWORD   <CJP,J=5200,D=1,CENA,CCMP>              ; 33
003653' 001400  130060
003654' 003551  000000                                MWORD   <JMAP,J=5100,D=1>                       ; 34
003655' 001000  000040
```

```
003656' 003651  000000          MWORD   <CJP,J=5100,D=1,CENA,CCMP>                      ; 35
003657' 001400  130060
003660' 003752  000000          MWORD   <JMAP,J=5200,D=1>                               ; 36
003661' 001000  000040



                        ;       Read a word from CBUS into CBIN

003662' 010001  010200          MWORD   <ADDR=100,JMAP,J=101,D=1,SELC,MGC=200>          ; 100
003663' 001000  004040
003664' 010101  030000          MWORD   <CJP,J=103,D=1,CENA,CCCA>                       ; 101
003665' 001400  000060
003666' 010201  010000          MWORD   <JMAP,J=101,D=1>                                ; 102
003667' 001000  000040
003670' 010300  000004          MWORD   <CONT,D=1,SELC,MGC=4>                           ; 103
003671' 001000  004340
003672' 010400  000200          MWORD   <CRTN,SDO,OR,B=3,D=2,SELM,MGC=200>              ; 104
003673' 732001  402240


                        ;       Ensure CBUS is stopped

003674' 020000  000120          MWORD   <ADDR=200,JZ,SELC,MGC=120>                      ; 200
003675' 000000  004000


                        ;       Initialization steps

003676' 500050  010100          MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>            ; 5000
003677' 000000  004040
003700' 500100  000020          MWORD   <CONT,SELM,MGC=20>                              ; 5001
003701' 000000  002340
003702' 500201  000000          MWORD   <CJS,J=100,D=1>                                 ; 5002
003703' 001000  000020
003704' 500300  002000          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=0>              ; 5003
003705' 401030  230340
003706' 500401  010000          MWORD   <CJS,J=101,D=1>                                 ; 5004
003707' 001000  000020
003710' 500500  002001          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=1>              ; 5005
003711' 401030  230340
003712' 500601  010000          MWORD   <CJS,J=101,D=1>                                 ; 5006
003713' 001000  000020
003714' 500700  002002          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=2>              ; 5007
003715' 401030  230340
003716' 501001  010000          MWORD   <CJS,J=101,D=1>                                 ; 5010
003717' 001000  000020
003720' 501100  002003          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=3>              ; 5011
003721' 401030  230340
003722' 501201  010000          MWORD   <CJS,J=101,D=1>                                 ; 5012
003723' 001000  000020
003724' 501300  002004          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=4>              ; 5013
003725' 401030  230340
003726' 501401  010000          MWORD   <CJS,J=101,D=1>                                 ; 5014
003727' 001000  000020
003730' 501500  002005          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=5>              ; 5015
003731' 401030  230340
003732' 501601  010000          MWORD   <CJS,J=101,D=1>                                 ; 5016
003733' 001000  000020
```

```
003734' 501700  002006          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=6>      ; 5017
003735' 401030  230340
003736' 502001  010000          MWORD   <CJS,J=101,D=1>                         ; 5020
003737' 001000  000020
003740' 502100  002007          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=7>      ; 5021
003741' 401030  230340
003742' 502200  000000          MWORD   <JZ>                                    ; 5022
003743' 000000  000000
003744' 000000  000000          MWORD   <ADDR=0,JZ>                             ; 5023
003745' 000000  000000

                        ; Subroutine - Passed - Write 1 to bit 12 of CSR register

003746' 510000  000000          MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>        ; 5100
003747' 742001  000340
003750' 510100  000000          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5101
003751' 302001  000740
003752' 510200  260000          MWORD   <LDCT,J=26>                             ; 5102
003753' 000000  000300
003754' 510351  030000          MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>      ; 5103
003755' 102021  000220
003756' 510400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5104
003757' 431020  005340
003760' 510551  070040          MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>  ; 5105
003761' 001400  015060
003762' 510651  050000          MWORD   <JMAP,J=5105,D=1>                       ; 5106
003763' 001000  000040
003764' 510700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5107
003765' 431010  005340
003766' 511051  100000          MWORD   <JMAP,J=5110,D=1>                       ; 5110
003767' 001000  000040

                        ; Subroutine - Failed - Write 1 to bit 13 of CSR register

003770' 520000  000000          MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>        ; 5200
003771' 742001  000340
003772' 520100  000000          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5201
003773' 302001  000740
003774' 520200  250000          MWORD   <LDCT,J=25>                             ; 5202
003775' 000000  000300
003776' 520352  030000          MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>      ; 5203
003777' 102021  000220
004000' 520400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5204
004001' 431020  005340
004002' 520552  070040          MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>  ; 5205
004003' 001400  015060
004004' 520652  050000          MWORD   <JMAP,J=5205,D=1>                       ; 5206
004005' 001000  000040
004006' 520700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5207
004007' 431010  005340
004010' 521052  100000          MWORD   <JMAP,J=5210,D=1>                       ; 5210
004011' 001000  000040
004012' 777777  777777          -1
```

```
                                     ;#*********************************************************************
                                     ;* Test 57 - Parity Predictor - PLIN to Fmtr/MPCOREDUMP=1
                                     ;
                                     ; Description:  Verify that the parity predictor works  correctly
                                     ;               when  moving  data  from  PLIN  to formatter with
                                     ;               'MPCOREDUMP' asserted.
                                     ;
                                     ; Procedure:    KL> Port Clear
                                     ;               KL> Set up a CCW list to transfer 10 words
                                     ;               KL> Start up port
                                     ;               UC> Read words from CBUS into CBIN to formatter
                                     ;                   then to local storage locations 0-11
                                     ;               UC> Initialize parity predictor with 'MPCLRPARCHK'
                                     ;
                                     ;               UC> Load PLOUT with word from local storage
                                     ;               UC> Load PLIN with word from PLOUT
                                     ;               UC> Assert 'MPPLINTOFMTR'
                                     ;               UC> Test leftmost bit of data word - this should
                                     ;                   be the setting of 'CCMVRPARCHK'.
                                     ;               UC> if correct - set bit 12 (DISABLE Complete)
                                     ;               UC> if not correct - set bit 13 (ENABLE Complete)
                                     ;               Read CSR and determine test disposition
                                     ;
                                     ;               Repeat the above steps for each of the 10 data
                                     ;               words.
                                     ;
                                     ; Failure:      ---
                                     ;#*********************************************************************

                                     ; Test data

004013' 254 00 0 00 004021'          TSTC57: JRST    TG57                         ; go start test
004014' 100604  000057                       CBUS!NDMP!NDCB!ZCBUS!57              ; test mask
004015' 004143' 005636'                      T57M,,[ASCIZ ^Parity Predictor - PLIN to Fmtr/MPCOREDUMP=1^]
004016' 005517' 005625'                      [EXP MLAST!C11],,[EXP C8,C12,C14,MLAST!C21]
004017' 000000  004352'                      TSTC60                               ; failure test table
004020' 777777  777777                       -1                                   ;   ...

                                     ; Start test

004021' 201 00 0 00 000000'          TG57:   MOVEI   ZA                           ; get address of module start
004022' 260 17 0 00 003501*                  GO      TRACE                        ; handle trace output
004023' 201 01 0 00 004143'                  MOVEI   1,T57M                       ; set up microcode address
004024' 260 17 0 00 003503*                  GO      TLOAD                        ; load/verify it
004025' 263 17 0 00 000000                   RTN                                  ; failed - exit test
```

```
DFPTAA
DFPTAA
 3877
 3878
 3879
 3880
 3881
 3882
 3883
 3884
 3885
 3886
 3887
 3888
 3889
 3890
 3891
 3892
 3893
 3894
 3895
 3896
 3897
 3898
 3899
 3900
 3901
 390
 390
 390
NO ERR

PROGR
CPU T

163P (

                                         ; Initialization

004026' 400 15 0 00 000000              TL57:   SETZ    ERFLG,                  ; clear error flag
004027' 260 17 0 00 003506*                      GO      IPACLR                  ; clear port
004030' 402 00 0 00 003507*                      SETZM   TSTSUB                  ; initialize subtest number
004031' 201 06 0 00 004043'                      MOVEI   6,TS57                  ; get test table address

                                         ; Loop on test execute table entries

004032' 260 17 0 00 003511*              TA57:   GO      TEXEC                   ; execute table entry
004033' 254 00 0 00 004042'                        JRST  TX57                    ; end of test execute table
004034' 254 00 0 00 004032'                        JRST  TA57                    ; keep looping after call
004035' 474 15 0 00 000000                         SETO  ERFLG,                  ; error occurred

                                         ; Handle error printouts and scope looping

004036' 027 00 0 00 004126'                      SCOPER  MA57                    ; print error message
004037' 254 00 0 00 004026'                      JRST    TL57                    ; loop on error
004040' 254 00 0 00 004042'                      JRST    TX57                    ; altmode exit
004041' 322 15 0 00 004032'                      JUMPE   ERFLG,TA57              ; do next test execute table entry

                                         ; End of test

004042' 263 17 0 00 000000               TX57:   RTN                             ; return

                                         ; Test Execute Table, as:  (CMD,parameters)

004043' 140000  004053'                  TS57:   TTABLE  (TCALL,TS57DA)          ; set up data transfer
004044' 140000  004104'                          TTABLE  (TCALL,TS57IN)          ; initialize counter
004045' 200000  004106'                  TS57A:  TTABLE  (TCALLC,TS57NE)         ; set up next test segment
004046' 400000  000010                           TTABLE  (TSSTAR,10)             ; start up port (read a word)
004047' 240002  000000                           TTABLE  (TCHECK,2)              ; check if completed
004050' 340000  000000                           TTABLE  (TEXIT)                 ; exit if error yet
004051' 300000  004045'                          TTABLE  (TJRST,TS57A)           ; loop till done
004052' 000000  000000                           TTABLE  (TLAST)                 ; end of table

                                         ; Set up data transfer

004053' 201 01 0 00 000200               TS57DA: MOVEI   1,200                   ; get start address
004054' 242 01 0 00 000001                       LSH     1,1                     ; position correctly
004055' 260 17 0 00 003553*                      GO      LDRAR                   ; load the register
004056' 201 01 0 00 000010                       MOVEI   1,MPRUN                 ; get start data
004057' 260 17 0 00 003561*                      GO      LDCSR                   ; start up the port
004060' 200 01 0 00 005103'                      MOVE    1,[CBUFSZ,,CBUF]        ; get pointer to control buffer
004061' 260 17 0 00 003540*                      GO      CHINIT                  ; initialize software
004062' 551 01 0 00 003541*                      HRRZI   1,BUFF                  ; buffer address
004063' 201 02 0 00 000012                       MOVEI   2,^D10                  ; word count
004064' 400 03 0 00 000000                       SETZ    3,                      ; clear zero fill specifier
004065' 260 17 0 00 003544*                      GO      GENCCW                  ; generate a CCW list
```

```
004066' 200 01 0 00 005647'                MOVE    1,[TS57BU,,BUFF]        ; copy the data for this test
                                                                        ;   into the data buffer
004067' 251 01 0 00 000000#                BLT     1,BUFF+^D9
004070' 400 01 0 00 000000                 SETZ    1,                      ; clear CSR data
004071' 260 17 0 00 004057*                GO      LDCSR                   ; write CSR register
004072' 201 01 0 00 005000                 MOVEI   1,5000                  ; get start address
004073' 242 01 0 00 000001                 LSH     1,1                     ; position correctly
004074' 260 17 0 00 004055*                GO      LDRAR                   ; load the register
004075' 201 01 0 00 000010                 MOVEI   1,MPRUN                 ; get start data
004076' 260 17 0 00 004071*                GO      LDCSR                   ; start up the port
004077' 201 00 0 00 000001                 MOVEI   1                       ; ensure the port has enough
004100' 260 17 0 00 003557*                GO      ODELAY                  ;   time to initialize
004101' 400 01 0 00 000000                 SETZ    1,                      ; clear CSR data
004102' 260 17 0 00 004076*                GO      LDCSR                   ; stop the port
004103' 263 17 0 00 000000                 RTN                             ; return

                                   ; Initialize test segment

004104' 476 00 0 00 005730'        TS57IN: SETOM   SAVSEG#                 ; initialize test segment
004105' 263 17 0 00 000000                 RTN                             ; return

                                   ; Point to next test segment

004106' 350 01 0 00 005730'        TS57NE: AOS     1,SAVSEG                ; point to next test segment
004107' 307 01 0 00 000011                 CAIG    1,^D9                   ; done yet?
004110' 350 00 0 17 000000                 AOS     (P)                     ; no - set up proper return
004111' 271 01 0 00 000010                 ADDI    1,10                    ; point to starting address
004112' 542 01 0 00 004046'                HRRM    1,TS57A+1               ; set up starting address
004113' 263 17 0 00 000000                 RTN                             ; return

                                   ; Data

004114' 400000  000000             TS57BU: 400000,,0
004115' 400000  000001                     400000,,1
004116' 400000  000020                     400000,,20
004117' 400000  000021                     400000,,21
004120' 400000  000040                     400000,,40
004121' 400000  000041                     400000,,41
004122' 400000  000100                     400000,,100
004123' 400000  000101                     400000,,101
004124' 400000  000200                     400000,,200
004125' 400000  000201                     400000,,201
```

```

                                        ; Error messages

004126' 140000  005524'                 MA57:   MSG!TXNOT![ASCIZ /Parity Predictor produced incorrect result:/]
004127' 240000  004131'                         CALL!TXNOT!MA57PN               ; print result
004130' 000000000000#                           LAST!CALL!TXALL!TTPNT           ; print test data

004131' 200 01 0 00 005730'             MA57PN: MOVE    1,SAVSEG                ; get test segment
004132' 037 00 0 00 005632'                     TMSGC   <  Data in PLIN:  >
004133' 550 00 0 01 004114'                     HRRZ    TS57BU(1)               ; get data word
004134' 037 03 0 00 000000                      PNT3                            ; print it
004135' 200 01 0 01 004114'                     MOVE    1,TS57BU(1)             ; get data word
004136' 607 01 0 00 400000                      TLNN    1,400000                ; should be cleared?
004137' 037 00 0 00 005541'                     TMSGC   <  'CCMVPARCHK' should not be asserted, but is>
004140' 603 01 0 00 400000                      TLNE    1,400000                ; should be set?
004141' 037 00 0 00 005553'                     TMSG    <  'CCMVPARCHK' should be asserted, but isn't>
004142' 263 17 0 00 000000                      RTN                             ; return

                                        ; Microcode

                                        ;       Read local storage location

004143' 001000  270000                  T57M:   MWORD   <ADDR=10,JMAP,J=27,SD0,OR,B=3,D=2,RDLM,MGC=0>
004144' 732001  620040
004145' 001100  300001                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=1>   ; 11
004146' 732001  620040
004147' 001200  300002                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=2>   ; 12
004150' 732001  620040
004151' 001300  300003                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=3>   ; 13
004152' 732001  620040
004153' 001400  300004                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=4>   ; 14
004154' 732001  620040
004155' 001500  300005                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=5>   ; 15
004156' 732001  620040
004157' 001600  300006                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=6>   ; 16
004160' 732001  620040
004161' 001700  300007                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=7>   ; 17
004162' 732001  620040
004163' 002000  300010                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=10>  ; 20
004164' 732001  620040
004165' 002100  300011                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=11>  ; 21
004166' 732001  620040


                                        ;       Do a test segment

004167' 002700  000030                          MWORD   <ADDR=27,CONT,D=1,SELM,MGC=30>          ; 27
004170' 001000  002340
004171' 003000  002040                          MWORD   <CONT,SOA,A=3,D=1,OENA,SELM,MGC=40>     ; 30
004172' 401030  002340
004173' 003100  000006                          MWORD   <CONT,SD0,D=1,SELP,MGC=6>               ; 31
004174' 701000  001340
004175' 003200  000020                          MWORD   <CONT,D=1,SELF,MGC=20>                  ; 32
004176' 001000  003340
004177' 003300  350000                          MWORD   <CJP,J=35,SOA,A=3,D=1,CENA,CCMB>        ; 32
004200' 401430  120060
```

```
 3447   004201' 003452  000000            MWORD   <CJP,J=5200,D=1,CENA,CCMP>           ; 33
 3448   004202' 001400  130060
 3449   004203' 003551  000000            MWORD   <JMAP,J=5100,D=1>                    ; 34
 3450   004204' 001000  000040
 3451   004205' 003651  000000            MWORD   <CJP,J=5100,D=1,CENA,CCMP>           ; 35
 3452   004206' 001400  130060
 3453   004207' 003752  000000            MWORD   <JMAP,J=5200,D=1>                    ; 36
 3454   004210' 001000  000040
 3455
 3456                             ;       Read a word from CBUS into CBIN
 3457
 3458   004211' 010001  010200            MWORD   <ADDR=100,JMAP,J=101,D=1,SELC,MGC=200>  ; 100
 3459   004212' 001000  004040
 3460   004213' 010101  030000            MWORD   <CJP,J=103,D=1,CENA,CCCA>            ; 101
 3461   004214' 001400  000060
 3462   004215' 010201  010000            MWORD   <JMAP,J=101,D=1>                     ; 102
 3463   004216' 001000  000040
 3464   004217' 010300  000004            MWORD   <CONT,D=1,SELC,MGC=4>                ; 103
 3465   004220' 001000  004340
 3466   004221' 010400  000200            MWORD   <CRTN,SDO,OR,B=3,D=2,SELM,MGC=200>   ; 104
 3467   004222' 732001  402240
 3468
 3469                             ;       Ensure CBUS is stopped
 3470
 3471   004223' 020000  000120            MWORD   <ADDR=200,JZ,SELC,MGC=120>           ; 200
 3472   004224' 000000  004000
 3473
 3474                             ;       Initialization steps
 3475
 3476   004225' 500050  010100            MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>  ; 5000
 3477   004226' 000000  004040
 3478   004227' 500100  000020            MWORD   <CONT,SELM,MGC=20>                   ; 5001
 3479   004230' 000000  002340
 3480   004231' 500201  000000            MWORD   <CJS,J=100,D=1>                      ; 5002
 3481   004232' 001000  000020
 3482   004233' 500300  002000            MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=0>   ; 5003
 3483   004234' 401030  230340
 3484   004235' 500401  010000            MWORD   <CJS,J=101,D=1>                      ; 5004
 3485   004236' 001000  000020
 3486   004237' 500500  002001            MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=1>   ; 5005
 3487   004240' 401030  230340
 3488   004241' 500601  010000            MWORD   <CJS,J=101,D=1>                      ; 5006
 3489   004242' 001000  000020
 3490   004243' 500700  002002            MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=2>   ; 5007
 3491   004244' 401030  230340
 3492   004245' 501001  010000            MWORD   <CJS,J=101,D=1>                      ; 5010
 3493   004246' 001000  000020
 3494   004247' 501100  002003            MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=3>   ; 5011
 3495   004250' 401030  230340
 3496   004251' 501201  010000            MWORD   <CJS,J=101,D=1>                      ; 5012
 3497   004252' 001000  000020
 3498   004253' 501300  002004            MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=4>   ; 5013
 3499   004254' 401030  230340
 3500   004255' 501401  010000            MWORD   <CJS,J=101,D=1>                      ; 5014
 3501   004256' 001000  000020
```

```
004257' 501500  002005          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=5>      ; 5015
004260' 401030  230340
004261' 501601  010000          MWORD   <CJS,J=101,D=1>                         ; 5016
004262' 001000  000020
004263' 501700  002006          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=6>      ; 5017
004264' 401030  230340
004265' 502001  010000          MWORD   <CJS,J=101,D=1>                         ; 5020
004266' 001000  000020
004267' 502100  002007          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=7>      ; 5021
004270' 401030  230340
004271' 502201  010000          MWORD   <CJS,J=101,D=1>                         ; 5022
004272' 001000  000020
004273' 502300  002010          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=10>     ; 5023
004274' 401030  230340
004275' 502401  010000          MWORD   <CJS,J=101,D=1>                         ; 5024
004276' 001000  000020
004277' 502500  002011          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=11>     ; 5025
004300' 401030  230340
004301' 502600  000000          MWORD   <JZ>                                    ; 5026
004302' 000000  000000
004303' 000000  000000          MWORD   <ADDR=0,JZ>                             ; 5027
004304' 000000  000000


                        ; Subroutine - Passed - Write 1 to bit 12 of CSR register

004305' 510000  000000          MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>        ; 5100
004306' 742001  000340
004307' 510100  000000          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5101
004310' 302001  000740
004311' 510200  260000          MWORD   <LDCT,J=26>                             ; 5102
004312' 000000  000300
004313' 510351  030000          MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>      ; 5103
004314' 102021  000220
004315' 510400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5104
004316' 431020  005340
004317' 510551  070040          MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>  ; 5105
004320' 001400  015060
004321' 510651  050000          MWORD   <JMAP,J=5105,D=1>                       ; 5106
004322' 001000  000040
004323' 510700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5107
004324' 431010  005340
004325' 511051  100000          MWORD   <JMAP,J=5110,D=1>                       ; 5110
004326' 001000  000040


                        ; Subroutine - Failed - Write 1 to bit 13 of CSR register

004327' 520000  000000          MWORD   <ADDR=5200,CONT,SDO,AND,B=2,D=2>        ; 5200
004330' 742001  000340
004331' 520100  000000          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>             ; 5201
004332' 302001  000740
004333' 520200  250000          MWORD   <LDCT,J=25>                             ; 5202
004334' 000000  000300
004335' 520352  030000          MWORD   <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>      ; 5203
004336' 102021  000220
004337' 520400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>   ; 5204
```

```
004340' 431020  005340
004341' 520552  070040          MWORD   <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>  ; 5205
004342' 001400  015060
004343' 520652  050000          MWORD   <JMAP,J=5205,D=1>                       ; 5206
004344' 001000  000040
004345' 520700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>      ; 5207
004346' 431010  005340
004347' 521052  100000          MWORD   <JMAP,J=5210,D=1>                       ; 5210
004350' 001000  000040
004351' 777777  777777          -1
```

```

                                           ;#*********************************************************************
                                           ;* Test 60 - Parity Predictor - Fmtr to CBOUT
                                           ;
                                           ; Description:  Verify that the parity predictor works  correctly
                                           ;               when moving data from formatter to CBOUT.
                                           ;
                                           ; Procedure:    KL> Port Clear
                                           ;               KL> Set up a CCW list to transfer 12 words
                                           ;               KL> Start up port
                                           ;               UC> Read words from CBUS into CBIN to formatter
                                           ;                   then to local storage locations 0-13
                                           ;               UC> Initialize parity predictor with 'MPCLRPARCHK'
                                           ;
                                           ;               UC> Load formatter with word from local storage
                                           ;               UC> Assert 'MPCBUSTOFMTR'
                                           ;               UC> Test leftmost bit of data word - this should
                                           ;                   be the setting of 'CCMVRPARCHK'.
                                           ;               UC> if correct - set bit 12 (DISABLE Complete)
                                           ;               UC> if not correct - set bit 13 (ENABLE Complete)
                                           ;               Read CSR and determine test disposition
                                           ;
                                           ;               Repeat the above steps for each of the 12 data
                                           ;               words.
                                           ;
                                           ; Failure:      ---
                                           ;#*********************************************************************

                                           ; Test data

004352' 254 00 0 00 004357'                TSTC60: JRST    TG60                    ; go start test
004353' 100604  000060                             CBUS!NDMP!NDCB!ZCBUS!60         ; test mask
004354' 004514' 005650'                            T60M,,[ASCIZ ^Parity Predictor - Fmtr to CBOUT^]
004355' 005517' 005625'                            [EXP MLAST!C11],,[EXP C8,C12,C14,MLAST!C21]
004356' 777777  777777                             -1                              ; failure test table

                                           ; Start test

004357' 201 00 0 00 000000'                TG60:   MOVEI   ZA                      ; get address of module start
004360' 260 17 0 00 004022*                        GO      TRACE                   ; handle trace output
004361' 201 01 0 00 004514'                        MOVEI   1,T60M                  ; set up microcode address
004362' 260 17 0 00 004024*                        GO      TLOAD                   ; load/verify it
004363' 263 17 0 00 000000                         RTN                             ; failed - exit test
```

```

                                 ; Initialization

 004364' 400 15 0 00 000000      TL60:    SETZ     ERFLG,                    ; clear error flag
 004365' 260 17 0 00 004027*              GO       IPACLR                    ; clear port
 004366' 402 00 0 00 004030*              SETZM    TSTSUB                    ; initialize subtest number
 004367' 201 06 0 00 004401'              MOVEI    6,TS60                    ; get test table address

                                 ; Loop on test execute table entries

 004370' 260 17 0 00 004032*     TA60:    GO       TEXEC                     ; execute table entry
 004371' 254 00 0 00 004400'               JRST    TX60                      ; end of test execute table
 004372' 254 00 0 00 004370'               JRST    TA60                      ; keep looping after call
 004373' 474 15 0 00 000000                SETO    ERFLG,                    ; error occurred

                                 ; Handle error printouts and scope looping

 004374' 027 00 0 00 004477'              SCOPER   MA60                      ; print error message
 004375' 254 00 0 00 004364'              JRST     TL60                      ; loop on error
 004376' 254 00 0 00 004400'              JRST     TX60                      ; altmode exit
 004377' 322 15 0 00 004370'              JUMPE    ERFLG,TA60                ; do next test execute table entry

                                 ; End of test

 004400' 263 17 0 00 000000      TX60:    RTN                                ; return

                                 ; Test Execute Table, as:  (CMD,parameters)

 004401' 140000  004411'         TS60:    TTABLE   (TCALL,TS60DA)            ; set up data transfer
 004402' 140000  004442'                  TTABLE   (TCALL,TS60IN)            ; initialize counter
 004403' 200000  004455'         TS60A:   TTABLE   (TCALLC,TS60NE)           ; set up next test segment
 004404' 400000  000010                   TTABLE   (TSSTAR,10)               ; start up port (read a word)
 004405' 240002  000000                   TTABLE   (TCHECK,2)                ; check if completed
 004406' 340000  000000                   TTABLE   (TEXIT)                   ; exit if error yet
 004407' 300000  004403'                  TTABLE   (TJRST,TS60A)             ; loop till done
 004410' 000000  000000                   TTABLE   (TLAST)                   ; end of table

                                 ; Set up data transfer

 004411' 201 01 0 00 000200      TS60DA: MOVEI     1,200                     ; get start address
 004412' 242 01 0 00 000001              LSH       1,1                       ; position correctly
 004413' 260 17 0 00 004074*             GO        LDRAR                     ; load the register
 004414' 201 01 0 00 000010              MOVEI     1,MPRUN                   ; get start data
 004415' 260 17 0 00 004102*             GO        LDCSR                     ; start up the port
 004416' 200 01 0 00 005103'             MOVE      1,[CBUFSZ,,CBUF]          ; get pointer to control buffer
 004417' 260 17 0 00 004061*             GO        CHINIT                    ; initialize software
 004420' 551 01 0 00 004062*             HRRZI     1,BUFF                    ; buffer address
 004421' 201 02 0 00 000014              MOVEI     2,^D12                    ; word count
 004422' 400 03 0 00 000000              SETZ      3,                        ; clear zero fill specifier
 004423' 260 17 0 00 004065*             GO        GENCCW                    ; generate a CCW list
```

```
004424' 200 01 0 00 005657'             MOVE    1,[TS60BU,,BUFF]        ; copy the data for this test
004425' 251 01 0 00 000000#             BLT     1,BUFF+^D11             ;   into the data buffer
004426' 400 01 0 00 000000              SETZ    1,                      ; clear CSR data
004427' 260 17 0 00 004415*             GO      LDCSR                   ; write CSR register
004430' 201 01 0 00 005000              MOVEI   1,5000                  ; get start address
004431' 242 01 0 00 000001              LSH     1,1                     ; position correctly
004432' 260 17 0 00 004413*             GO      LDRAR                   ; load the register
004433' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get start data
004434' 260 17 0 00 004427*             GO      LDCSR                   ; start up the port
004435' 201 00 0 00 000001              MOVEI   1                       ; ensure the port has enough
004436' 260 17 0 00 004100*             GO      ODELAY                  ;   time to initialize
004437' 400 01 0 00 000000              SETZ    1,                      ; clear CSR data
004440' 260 17 0 00 004434*             GO      LDCSR                   ; stop the port
004441' 263 17 0 00 000000              RTN                             ; return

                                ; Initialize test segment

004442' 476 00 0 00 005730'     TS60IN: SETOM   SAVSEG#                 ; initialize test segment
004443' 400 01 0 00 000000              SETZ    1,                      ; clear CSR data
004444' 260 17 0 00 004440*             GO      LDCSR                   ; write CSR register
004445' 201 01 0 00 000200              MOVEI   1,200                   ; get start address
004446' 242 01 0 00 000001              LSH     1,1                     ; position correctly
004447' 260 17 0 00 004432*             GO      LDRAR                   ; load the register
004450' 201 01 0 00 000010              MOVEI   1,MPRUN                 ; get start data
004451' 260 17 0 00 004444*             GO      LDCSR                   ; start up the port
004452' 400 01 0 00 000000              SETZ    1,                      ; clear CSR data
004453' 260 17 0 00 004451*             GO      LDCSR                   ; stop the port
004454' 263 17 0 00 000000              RTN                             ; return

                                ; Point to next test segment

004455' 350 01 0 00 005730'     TS60NE: AOS     1,SAVSEG                ; point to next test segment
004456' 307 01 0 00 000013              CAIG    1,^D11                  ; done yet?
004457' 350 00 0 17 000000              AOS     (P)                     ; no - set up proper return
004460' 271 01 0 00 000010              ADDI    1,10                    ; point to starting address
004461' 542 01 0 00 004404'             HRRM    1,TS60A+1               ; set up starting address
004462' 263 17 0 00 000000              RTN                             ; return

                                ; Data

004463' 400000  000000          TS60BU: 400000,,0
004464' 000001  000000                  1,,0
004465' 400000  000000                  400000,,0
004466' 400000  000020                  400000,,20
004467' 000000  000020                  0,,20
004470' 400001  000020                  400001,,20
004471' 400001  000000                  400001,,0
004472' 400001  000000                  400001,,0
004473' 400001  000000                  400001,,0
004474' 400003  000020                  400003,,20
004475' 400003  000020                  400003,,20
004476' 400003  000020                  400003,,20
```

```

                                        ; Error messages

004477' 140000  005524'                 MA60:   MSG!TXNOT![ASCIZ /Parity Predictor produced incorrect result:/]
004500' 240000  004502'                         CALL!TXNOT!MA60PN               ; print result
004501' 000000000000#                           LAST!CALL!TXALL!TTPNT           ; print test data

004502' 200 01 0 00 005730'             MA60PN: MOVE    1,SAVSEG                ; get test segment
004503' 037 00 0 00 005660'                     TMSGC   <  Data in Fmtr:  >
004504' 550 00 0 01 004463'                     HRRZ    TS60BU(1)               ; get data word
004505' 037 16 0 00 000003                      PNTOCS                          ; print it
004506' 200 01 0 01 004463'                     MOVE    1,TS60BU(1)             ; get data word
004507' 607 01 0 00 400000                      TLNN    1,400000                ; should be cleared?
004510' 037 00 0 00 005541'                     TMSGC   <  'CCMVPARCHK' should not be asserted, but is>
004511' 603 01 0 00 400000                      TLNE    1,400000                ; should be set?
004512' 037 00 0 00 005553'                     TMSGC   <  'CCMVPARCHK' should be asserted, but isn't>
004513' 263 17 0 00 000000                      RTN                             ; return

                                        ; Microcode

                                        ;       Read local storage location

004514' 001000  270000                  T60M:   MWORD   <ADDR=10,JMAP,J=27,SD0,OR,B=3,D=2,RDLM,MGC=0>
004515' 732001  620040
004516' 001100  300001                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=1>   ; 11
004517' 732001  620040
004520' 001200  300002                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=2>   ; 12
004521' 732001  620040
004522' 001300  300003                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=3>   ; 13
004523' 732001  620040
004524' 001400  300004                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=4>   ; 14
004525' 732001  620040
004526' 001500  300005                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=5>   ; 15
004527' 732001  620040
004530' 001600  300006                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=6>   ; 16
004531' 732001  620040
004532' 001700  300007                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=7>   ; 17
004533' 732001  620040
004534' 002000  300010                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=10>  ; 20
004535' 732001  620040
004536' 002100  300011                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=11>  ; 21
004537' 732001  620040
004540' 002200  300012                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=12>  ; 22
004541' 732001  620040
004542' 002300  300013                          MWORD   <JMAP,J=30,SD0,OR,B=3,D=2,RDLM,MGC=13>  ; 23
004543' 732001  620040

                                        ;       Do a test segment

004544' 002700  000030                          MWORD   <ADDR=27,CONT,D=1,SELM,MGC=30>          ; 27
004545' 001000  002340
004546' 003000  002040                          MWORD   <CONT,SOA,A=3,D=1,OENA,SELF,MGC=40>     ; 30
004547' 401030  003340
004550' 003100  000002                          MWORD   <CONT,D=1,SELC,MGC=2>                   ; 31
004551' 001000  004340
```

```
 3767  004552' 003200  350000                  MWORD   <CJP,J=35,SOA,A=3,D=1,CENA,CCMB>        ; 32
 3768  004553' 401430  120060
 3769  004554' 003352  000000                  MWORD   <CJP,J=5200,D=1,CENA,CCMP>              ; 33
 3770  004555' 001400  130060
 3771  004556' 003451  000000                  MWORD   <JMAP,J=5100,D=1>                       ; 34
 3772  004557' 001000  000040
 3773  004560' 003551  000000                  MWORD   <CJP,J=5100,D=1,CENA,CCMP>              ; 35
 3774  004561' 001400  130060
 3775  004562' 003652  000000                  MWORD   <JMAP,J=5200,D=1>                       ; 36
 3776  004563' 001000  000040
 3777
 3778                                  ;       Read a word from CBUS into CBIN
 3779
 3780  004564' 010001  010200                  MWORD   <ADDR=100,JMAP,J=101,D=1,SELC,MGC=200>  ; 100
 3781  004565' 001000  004040
 3782  004566' 010101  030000                  MWORD   <CJP,J=103,D=1,CENA,CCCA>               ; 101
 3783  004567' 001400  000060
 3784  004570' 010201  010000                  MWORD   <JMAP,J=101,D=1>                        ; 102
 3785  004571' 001000  000040
 3786  004572' 010300  000004                  MWORD   <CONT,D=1,SELC,MGC=4>                   ; 103
 3787  004573' 001000  004340
 3788  004574' 010400  000200                  MWORD   <CRTN,SDO,OR,B=3,D=2,SELM,MGC=200>      ; 104
 3789  004575' 732001  402240
 3790
 3791                                  ;       Ensure CBUS is stopped
 3792
 3793  004576' 020000  000120                  MWORD   <ADDR=200,JZ,SELC,MGC=120>              ; 200
 3794  004577' 000000  004000
 3795
 3796                                  ;       Initialization steps
 3797
 3798  004600' 500050  010100                  MWORD   <ADDR=5000,JMAP,J=5001,SELC,MGC=100>    ; 5000
 3799  004601' 000000  004040
 3800  004602' 500100  000020                  MWORD   <CONT,SELM,MGC=20>                      ; 5001
 3801  004603' 000000  002340
 3802  004604' 500201  000000                  MWORD   <CJS,J=100,D=1>                         ; 5002
 3803  004605' 001000  000020
 3804  004606' 500300  002000                  MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=0>      ; 5003
 3805  004607' 401030  230340
 3806  004610' 500401  010000                  MWORD   <CJS,J=101,D=1>                         ; 5004
 3807  004611' 001000  000020
 3808  004612' 500500  002001                  MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=1>      ; 5005
 3809  004613' 401030  230340
 3810  004614' 500601  010000                  MWORD   <CJS,J=101,D=1>                         ; 5006
 3811  004615' 001000  000020
 3812  004616' 500700  002002                  MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=2>      ; 5007
 3813  004617' 401030  230340
 3814  004620' 501001  010000                  MWORD   <CJS,J=101,D=1>                         ; 5010
 3815  004621' 001000  000020
 3816  004622' 501100  002003                  MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=3>      ; 5011
 3817  004623' 401030  230340
 3818  004624' 501201  010000                  MWORD   <CJS,J=101,D=1>                         ; 5012
 3819  004625' 001000  000020
 3820  004626' 501300  002004                  MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=4>      ; 5013
 3821  004627' 401030  230340
```

```
3822  004630' 501401  010000          MWORD   <CJS,J=101,D=1>                             ; 5014
3823  004631' 001000  000020
3824  004632' 501500  002005          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=5>          ; 5015
3825  004633' 401030  230340
3826  004634' 501601  010000          MWORD   <CJS,J=101,D=1>                             ; 5016
3827  004635' 001000  000020
3828  004636' 501700  002006          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=6>          ; 5017
3829  004637' 401030  230340
3830  004640' 502001  010000          MWORD   <CJS,J=101,D=1>                             ; 5020
3831  004641' 001000  000020
3832  004642' 502100  002007          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=7>          ; 5021
3833  004643' 401030  230340
3834  004644' 502201  010000          MWORD   <CJS,J=101,D=1>                             ; 5022
3835  004645' 001000  000020
3836  004646' 502300  002010          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=10>         ; 5023
3837  004647' 401030  230340
3838  004650' 502401  010000          MWORD   <CJS,J=101,D=1>                             ; 5024
3839  004651' 001000  000020
3840  004652' 502500  002011          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=11>         ; 5025
3841  004653' 401030  230340
3842  004654' 502601  010000          MWORD   <CJS,J=101,D=1>                             ; 5026
3843  004655' 001000  000020
3844  004656' 502700  002012          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=12>         ; 5027
3845  004657' 401030  230340
3846  004660' 503001  010000          MWORD   <CJS,J=101,D=1>                             ; 5030
3847  004661' 001000  000020
3848  004662' 503100  002013          MWORD   <CONT,SOA,A=3,D=1,OENA,LDLM,MGC=13>         ; 5031
3849  004663' 401030  230340
3850  004664' 503200  000000          MWORD   <JZ>                                        ; 5032
3851  004665' 000000  000000
3852  004666' 000000  000000          MWORD   <ADDR=0,JZ>                                 ; 5023
3853  004667' 000000  000000
3854
3855                          ; Subroutine - Passed - Write 1 to bit 12 of CSR register
3856
3857  004670' 510000  000000          MWORD   <ADDR=5100,CONT,SDO,AND,B=2,D=2>            ; 5100
3858  004671' 742001  000340
3859  004672' 510100  000000          MWORD   <CONT,SOB,PLUS,B=2,D=2,CRY>                 ; 5101
3860  004673' 302001  000740
3861  004674' 510200  260000          MWORD   <LDCT,J=26>                                 ; 5102
3862  004675' 000000  000300
3863  004676' 510351  030000          MWORD   <RPCT,J=5103,SAB,PLUS,A=2,B=2,D=2>          ; 5103
3864  004677' 102021  000220
3865  004700' 510400  002004          MWORD   <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>       ; 5104
3866  004701' 431020  005340
3867  004702' 510551  070040          MWORD   <CJP,J=5107,CENA,CCGC,D=1,SELE,MGC=40>      ; 5105
3868  004703' 001400  015060
3869  004704' 510651  050000          MWORD   <JMAP,J=5105,D=1>                           ; 5106
3870  004705' 001000  000040
3871  004706' 510700  000200          MWORD   <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>          ; 5107
3872  004707' 431010  005340
3873  004710' 511051  100000          MWORD   <JMAP,J=5110,D=1>                           ; 5110
3874  004711' 001000  000040
3875
3876                          ; Subroutine - Failed - Write 1 to bit 13 of CSR register
```

```
 3877
 3878  004712' 520000  000000          MWORD  <ADDR=5200,CONT,SDO,AND,B=2,D=2>          ; 5200
 3879  004713' 742001  000340
 3880  004714' 520100  000000          MWORD  <CONT,SOB,PLUS,B=2,D=2,CRY>               ; 5201
 3881  004715' 302001  000740
 3882  004716' 520200  250000          MWORD  <LDCT,J=25>                               ; 5202
 3883  004717' 000000  000300
 3884  004720' 520352  030000          MWORD  <RPCT,J=5203,SAB,PLUS,A=2,B=2,D=2>        ; 5203
 3885  004721' 102021  000220
 3886  004722' 520400  002004          MWORD  <CONT,SOA,A=2,OR,D=1,OENA,SELE,MGC=4>     ; 5204
 3887  004723' 431020  005340
 3888  004724' 520552  070040          MWORD  <CJP,J=5207,CENA,CCGC,D=1,SELE,MGC=40>    ; 5205
 3889  004725' 001400  015060
 3890  004726' 520652  050000          MWORD  <JMAP,J=5205,D=1>                         ; 5206
 3891  004727' 001000  000040
 3892  004730' 520700  000200          MWORD  <CONT,SOA,A=1,OR,D=1,SELE,MGC=200>        ; 5207
 3893  004731' 431010  005340
 3894  004732' 521052  100000          MWORD  <JMAP,J=5210,D=1>                         ; 5210
 3895  004733' 001000  000040
 3896  004734' 777777  777777          -1
 3897
 3898                          ;#*********************************************************************
 3899                          ; End of CBUS Module Tests
 3900                          ;#*********************************************************************
 3901
 3902                                  XLIST
 3903
 3904
NO ERRORS DETECTED

PROGRAM BREAK IS 005731
CPU TIME USED 04:22.002

163P CORE USED
```

```
APR              000000   spd   MA46             001236'        SCOSW            002021'  ext   TG43             000512'
BBPNT            000000   ext   MA46CO           001252'        SINCYC           020000   spd   TG44             000603'
BEXEC            002157'  ext   MA46CH           001241'        SSCALL           000003   spd   TG45             001014'
BUFCOM           001263'  ext   MA47             001431'        SSLAST           000000   spd   TG46             001200'
BUFF             005657'  ext   MA47CO           001446'        SSSTRT           000001   spd   TG47             001334'
C1               000054   spd   MA47CH           001434'        T40M             000052'        TG47A            001342'
C11              000066   spd   MA50             001631'        T41M             000272'        TG50             001560'
C12              000067   spd   MA50PN           001632'        T42M             000454'        TG51             001676'
C14              000071   spd   MA51             002017'        T43M             000543'        TG52             002146'
C17              000074   spd   MA51LA           002053'        T44M             000654'        TG53             002225'
C18              000075   spd   MA51LG           002037'        T45M             001111'        TG54             002611'
C20              000077   spd   MA51PN           002021'        T46M             001265'        TG55             003160'
C21              000100   spd   MA52             002176'        T47M             001470'        TG56             003500'
C22              000101   spd   MA53             002430'        T50M             001657'        TG57             004021'
C23              000102   spd   MA53PN           002433'        T50SS            001616'        TG60             004357'
C24              000103   spd   MA54             003000'        T51M             002116'        TJRST            000006   spd
C4               000057   spd   MA54PN           003003'        T52M             002200'        TL40             000016'
C7               000062   spd   MA55             003263'        T53M             002445'        TL41             000200'
C8               000063   spd   MA55PN           003266'        T54M             003015'        TL42             000430'
C9               000064   spd   MA56             003603'        T55M             003300'        TL43             000517'
CALL     200000  000000   spd   MA56PN           003606'        T56M             003620'        TL44             000610'
CBUF             005103'  ext   MA57             004126'        T57M             004143'        TL45             001021'
CBUFSZ           000315   spd   MA57PN           004131'        T60M             004514'        TL46             001205'
CBUS     100000  000000   spd   MA60             004477'        TA40             000022'        TL47             001347'
CCALL            000005   spd   MA60PN           004502'        TA41             000204'        TL51             001705'
CCPNT            000000   ext   MLAST    400000  000000   spd   TA42             000434'        TL52             002153'
CEBUFA           005250'  ext   MPRUN            000010   spd   TA43             000523'        TL53             002232'
CEXEC            001353'  ext   MSG      100000  000000   spd   TA44             000614'        TL54             002616'
CHINIT           004417'  ext   NDCB     000200  000000   spd   TA45             001024'        TL55             003165'
CLAST            000000   spd   NDMP     000400  000000   spd   TA46             001210'        TL56             003505'
CSETRD           000001   spd   ODELAY           004436'  ext   TA47             001352'        TL57             004026'
CSETWR           000002   spd   P                000017         TA50             001567'        TL60             004364'
CSTART           000004   spd   PATBUF           000000   ext   TA51             001710'        TLAST            000000   spd
CSTATF           000613'  ext   PDEC     006000  000000         TA52             002157'        TLOAD            004362'  ext
CWAIT            000007   spd   PI               000004   spd   TA53             002236'        TRACE            004360'  ext
ERFLG            000015         PNT3     037140  000000         TA54             002622'        TS40             000034'
GENCCW           004423'  ext   PNTDEC   037640  000000         TA55             003171'        TS41             000216'
GETLOG           001751'  ext   PNTHW    037540  000000         TA56             003511'        TS42             000445'
GO       260740  000000         PNTMSG   037000  000000         TA57             004032'        TS43             000534'
IPACLR           004365'  ext   PNTOCS   037700  000003         TA60             004370'        TS44             000626'
LAST     010000  000000   spd   RDEBUF           002644'  ext   TB40             000023'        TS44DA           000633'
LDCSR            004453'  ext   RTN      263740  000000         TB41             000205'        TS45             001036'
LDRAR            004447'  ext   SAVAPR           005715'        TB44             000615'        TS45CH           001043'
LOGBUF           000000   ext   SAVCN1           005716'        TB45             001025'        TS46             001222'
MA40             000043'        SAVCN2           005717'        TB46             001211'        TS46CH           001227'
MA40L            000050'        SAVCN3           005720'        TB47             001353'        TS47             001373'
MA40LP           000045'        SAVCN4           005721'        TB51             001711'        TS47CH           001416'
MA41             000222'        SAVDA1           005722'        TCALL            000003   spd   TS47DO           001404'
MA41LP           000224'        SAVDAT           005723'        TCALLC           000004   spd   TS47DA           001401'
MA42             000452'        SAVNUM           005724'        TCHECK           000005   spd   TS51             001722'
MA43             000541'        SAVR11           005725'        TEXEC            004370'  ext   TS51CO           001766'
MA44             000652'        SAVR12           005726'        TEXIT            000007   spd   TS51CD           002013'
MA45             001052'        SAVR13           005727'        TG40             000011'        TS51CH           001747'
MA45CO           001067'        SAVSEG           005730'        TG41             000173'        TS51IN           001727'
MA45CH           001055'        SCOPER   027000  000000         TG42             000423'        TS51TB           002003'
```

```
TS51TC          002007'       TUSER    001000 000000  spd   .LMGC           000001  spd   .MSELF          000007  spd
TS52            002170'       TX40            000033'       .LOENA          002000  spd   .MSELM          000007  spd
TS53            002267'       TX41            000215'       .LOR            000000  spd   .MSELP          000007  spd
TS53BU          002320'       TX42            000444'       .LPLUS          000000  spd   .MSKCN          000037  spd
TS53DA          002273'       TX43            000533'       .LRDLM          000000  spd   .MSMIN          000007  spd
TS54            002653'       TX44            000625'       .LRMIN          000000  spd   .MTIM           000001  spd
TS54BU          002704'       TX45            001035'       .LRPCT          000000  spd   .MXNOR          000007  spd
TS54DA          002657'       TX46            001221'       .LSOA           000000  spd   .RA      000010 000000  spd
TS55            003202'       TX47            001363'       .LSOB           000000  spd   .RAND    040000 000000  spd
TS55A           003204'       TX50            001615'       .LSAB           000000  spd   .RB             400000  spd
TS55BU          003253'       TX51            001721'       .LSDO           000000  spd   .RCCCA          000000  spd
TS55DA          003212'       TX52            002167'       .LSELC          000000  spd   .RCCCP          140000  spd
TS55IN          003243'       TX53            002266'       .LSELE          000000  spd   .RCCFZ          020000  spd
TS55NE          003245'       TX54            002652'       .LSELF          000000  spd   .RCCGC          010000  spd
TS56            003522'       TX55            003201'       .LSELM          000000  spd   .RCCLW          170000  spd
TS56A           003524'       TX56            003521'       .LSELP          000000  spd   .RCCMB          120000  spd
TS56BU          003573'       TX57            004042'       .LSKCN          000000  spd   .RCCMP          130000  spd
TS56DA          003532'       TX60            004400'       .LSMIN          000000  spd   .RCENA   000400 000000  spd
TS56IN          003563'       TXALL    060000 000000  spd   .LTIM           000000  spd   .RCJP           000060  spd
TS56NE          003565'       TXNOT    040000 000000  spd   .LXNOR          000000  spd   .RCJS           000020  spd
TS57            004043'       TXTINH          000200  spd   .MA             000017  spd   .RCONT          000340  spd
TS57A           004045'       UDEBUG          005300' ext   .MAND           000007  spd   .RCRTN          000240  spd
TS57BU          004114'       USER            030037        .MB             000017  spd   .RCRY           000400  spd
TS57DA          004053'       ZA              000000'       .MCCCA          000037  spd   .RD      001000 000000  spd
TS57IN          004104'       ZCBUS    000004 000000  spd   .MCCCP          000037  spd   .RJ             000000  spd
TS57NE          004106'       $ARG2           000001        .MCCFZ          000037  spd   .RJMAP          000040  spd
TS60            004401'       $B              000044        .MCCGC          000037  spd   .RJZ            000000  spd
TS60A           004403'       $CHR            000044        .MCCLW          000037  spd   .RLDCT          000300  spd
TS60BU          004463'       $GARG           000001        .MCCMB          000037  spd   .RLDLM          230000  spd
TS60DA          004411'       %ADDR           005211  spd   .MCCMP          000037  spd   .RMGC           000000  spd
TS60IN          004442'       %ML      521052 100000  spd   .MCENA          000001  spd   .ROENA          000000  spd
TS60NE          004455'       %MR      001000 000040  spd   .MCJP           000017  spd   .ROR     030000 000000  spd
TSSTAR          000010  spd   .LA             000000  spd   .MCJS           000017  spd   .RPLUS          000000  spd
TSTART          000001  spd   .LADDR   000100 000000  spd   .MCONT          000017  spd   .RRDLM          220000  spd
TSTC40          000000' ent   .LAND           000000  spd   .MCRTN          000017  spd   .RRMIN   020000 000000  spd
TSTC41          000163' ent   .LB             000000  spd   .MCRY           000001  spd   .RRPCT          000220  spd
TSTC42          000413' ent   .LCCCA          000000  spd   .MD             000007  spd   .RSOA    400000 000000  spd
TSTC43          000503' ent   .LCCCP          000000  spd   .MJ             007777  spd   .RSOB    300000 000000  spd
TSTC44          000572' ent   .LCCFZ          000000  spd   .MJMAP          000017  spd   .RSAB    100000 000000  spd
TSTC45          001005' ent   .LCCGC          000000  spd   .MJZ            000017  spd   .RSDO    700000 000000  spd
TSTC46          001172' ent   .LCCLW          000000  spd   .MLDCT          000017  spd   .RSELC          004000  spd
TSTC47          001326' ent   .LCCMB          000000  spd   .MLDLM          000037  spd   .RSELE          005000  spd
TSTC50          001553' ent   .LCCMP          000000  spd   .MMGC           001777  spd   .RSELF          003000  spd
TSTC51          001670' ent   .LCENA          000000  spd   .MOENA          000001  spd   .RSELM          002000  spd
TSTC52          002141' ent   .LCJP           000000  spd   .MOR            000007  spd   .RSELP          001000  spd
TSTC53          002213' ent   .LCJS           000000  spd   .MPLUS          000007  spd   .RSKCN          240000  spd
TSTC54          002600' ent   .LCONT          000000  spd   .MRDLM          000037  spd   .RSMIN   010000 000000  spd
TSTC55          003150' ent   .LCRTN          000000  spd   .MRMIN          000007  spd   .RTIM           000010  spd
TSTC56          003471' ent   .LCRY           000000  spd   .MRPCT          000017  spd   .RXNOR   070000 000000  spd
TSTC57          004013' ent   .LD             000000  spd   .MSOA           000007  spd
TSTC60          004352' ent   .LJ             010000  spd   .MSOB           000007  spd
TSTEBF          200000  spd   .LJMAP          000000  spd   .MSAB           000007  spd
TSTFLG          005301' ext   .LJZ            000000  spd   .MSDO           000007  spd
TSTSUB          004366' ext   .LLDCT          000000  spd   .MSELC          000007  spd
TTPNT           000000  ext   .LLDLM          000000  spd   .MSELE          000007  spd
```

```
APR        1332   1334   1366   1370
BBPNT       16#    547    658   1971
BEXEC       16#    520    631   1942
BUFCOM      24#   1226   1233
BUFF        15#    781    782    785   1384   1385   1386   1394   1400   1751   1756   1777   1803   2103
           2107   2108   2445   2449   2450   2751   2755   2756   3049   3053   3054   3347   3351   3352
           3656   3660   3661
C1         1567   1695
C11        2019   2362   2692   2991   3290   3600
C12        2019   2362   2692   2991   3290   3600
C14        1922   2019   2362   2692   2991   3290   3600
C17         948   1149   1319
C18         261    724
C20         724   1922
C21        2019   2362   2692   2991   3290   3600
C22         261    497    609
C23          70
C24         261
C4         1922
C7          948   1149   1319   1922
C8           70    497    609    724   2991   3290   3600
C9          948   1149   1319
CALL        128    129    310    311    547    658    794   1007   1008   1207   1208   1415   1416   1636
           1809   1971   2195   2196   2524   2525   2798   2799   3096   3097   3396   3397   3716   3717
CBUF        15#    783   1749   1801   1802   2101   2443   2749   3047   3345   3654
CBUFSZ      783   1749   2101   2443   2749   3047   3345   3654
CBUS         68    259    495    607    722    946   1147   1317   1565   1693   1920   2017   2360   2690
           2989   3288   3598
CCALL       991    992   1191   1192   1376   1377   1379   1380
CCPNT       16#   1008   1208   1416
CEBUFA      16#    997   1013   1016   1020   1024   1197   1213   1219   1405   1421   1424   1428   1432
CEXEC       16#    970   1170   1350
CHINIT      20#    784   1750   2102   2444   2750   3048   3346   3655
CLAST       992    993   1192   1193   1380   1381
CSETRD     1188   1189
CSETWR      988    989   1375   1376
CSTART      989    990   1189   1190   1377   1378
CSTATF      16#    744
CWAIT       990    991   1190   1191   1378   1379
ERFLG        87     98    105    277    288    295    513    523    530    624    634    641    741    752
            759    963    973    980    999   1012   1043   1163   1173   1180   1199   1212   1228   1343
           1353   1360   1407   1420   1451   1586   1602   1608   1711   1721   1728   1786   1789   1792
           1793   1935   1945   1952   2037   2047   2074   2379   2389   2416   2708   2718   2725   3006
           3016   3023   3304   3314   3321   3613   3623   3630
GENCCW      20#    788   1754   2106   2448   2754   3052   3350   3659
GETLOG      20#   1765
IPACLR      20#     88     94    278    284    514    625    748    969   1169   1349   1622   1717   1936
           1962   2038   2380   2709   3007   3305   3614
LAST        129    311    547    658    794   1008   1208   1416   1636   1809   1971   2196   2525   2799
           3097   3397   3717
LDCSR       20#    314    319    324    329    334    339    779    780   1031   1036   1038   1042   1439
           1444   1446   1450   1626   1630   1748   1758   2055   2065   2091   2096   2100   2397   2407
           2433   2438   2442   2748   2758   2763   2767   3046   3056   3061   3065   3344   3354   3359
```

%ML

```
             3363     3653     3663     3668     3672     3679     3684     3686                                                         SEQ 1968
LDRAR         20#      317      327      337      777     1034     1442     1624     1746     2053     2063     2094     2395     2405
             2436     2746     2761     3044     3059     3342     3357     3651     3666     3682
LOGBUF        20#     1766
MA40          102      128#
MA40L         132      135#
MA40LP        128      131#
MA41          292      310#
MA41LP        310      313#
MA42          527      546#
MA43          638      657#
MA44          756      793#
MA45          977     1006#
MA45CO       1012     1022     1030#
MA45CH        996     1007     1012#    1015
MA46         1177     1206#
MA46CO       1212     1215     1224#
MA46CH       1196     1207     1212#
MA47         1357     1414#
MA47CO       1420     1430     1438#
MA47CH       1404     1415     1420#    1423
MA50         1612     1636#
MA50PN       1636     1638#
MA51         1725     1808#
MA51LA       1829     1841#
MA51LG       1817     1823     1828#
MA51PN       1809     1811#
MA52         1949     1969#
MA53         2071     2194#
MA53PN       2195     2198#
MA54         2413     2523#
MA54PN       2524     2527#
MA55         2722     2797#
MA55PN       2798     2801#
MA56         3020     3095#
MA56PN       3096     3099#
MA57         3318     3395#
MA57PN       3396     3399#
MA60         3627     3715#
MA60PN       3716     3719#
MLAST          70      261      497      609      724      948     1149     1319     1567     1695     1922     2019     2362     2692
             2991     3290     3600
MPRUN         318      328      338      778     1035     1443     1625     1747     2054     2064     2095     2396     2406     2437
             2747     2762     3045     3060     3343     3358     3652     3667     3683
MSG           546      657      793     1006     1206     1414     1808     1969     2194     2523     2797     3095     3395     3715
NDCB           68      259      495      607      722      946     1147     1317     1565     1693     1920     2017     2360     2690
             2989     3288     3598
NDMP           68      259      495      607      722      946     1147     1317     1565     1693     1920     2017     2360     2690
             2989     3288     3598
ODELAY        15#     1628     1764     2098     2440     2765     3063     3361     3670
P            1000     1001     1200     1201     1408     1409     1794     1795     2779     3077     3375     3693
PATBUF        24#
PI           1389     1392
```

```
RDEBUF      20#     320      330      340     1039     1447     1631     2056     2066     2398     2408
SAVAPR    1325#    1325     1333     1364     1367
SAVCN1    1589#    1589     1597     1603     1643
SAVCN2    1593#    1593     1601     1607     1655
SAVCN3    1600#    1600     1651
SAVCN4    1606#    1606     1647
SAVDA1    1040#    1040     1045     1448#    1448     1453
SAVDAT    2067#    2067     2409
SAVNUM    1227#    1227
SAVR11     321#     321      342
SAVR12     331#     331      348
SAVR13     341#     341      351
SAVSEG    2057     2109#    2109     2198     2399     2451#    2451     2527     2772#    2772     2777     2801     3070#    3070
          3075     3099     3368#    3368     3373     3399     3677#    3677     3691     3719
SCOSW       24#    1639     1811
SINCYC     318      328      338      778     2054     2064     2396     2406
SSCALL    1962     1963
SSLAST     542      543      653      654     1965     1966
SSSTRT     538      539      540      541      542      649      650      651      652      653     1960     1961     1962     1963
          1964     1965
T40M        69       81      142#
T41M       260      271      360#
T42M       496      507      551#
T43M       608      618      662#
T44M       723      735      800#
T45M       947      957     1051#
T46M      1148     1157     1240#
T47M      1318     1337     1459#
T50M      1566     1580     1662#
T50SS     1588     1592     1622#
T51M      1694     1703     1879#
T52M      1921     1929     1975#
T53M      2018     2031     2213#
T54M      2361     2373     2542#
T55M      2691     2702     2816#
T56M      2990     3000     3114#
T57M      3289     3298     3414#
T60M      3599     3607     3734#
TA40        94#     105
TA41       284#     295
TA42       520#     522      530
TA43       631#     633      641
TA44       748#     759
TA45       969#     980
TA46      1169#    1180
TA47      1349#    1360
TA50      1586#    1613
TA51      1717#    1728
TA52      1942#    1944     1952
TA53      2044#    2046     2074
TA54      2386#    2388     2416
TA55      2715#    2717     2725
TA56      3013#    3015     3023
```

```
TA57      3311#   3313    3321
TA60      3620#   3622    3630
TB40        95#     97
TB41       285#    287
TB44       749#    751
TB45       970#    972
TB46      1170#   1172
TB47      1350#   1352
TB51      1718#   1720
TCALL      767     768    1736    1737    1738    1739    2082    2083    2424    2425    2733    2734    2735    3031
          3032    3033    3329    3330    3331    3638    3639    3640
TCALLC    2735    2736    3033    3034    3331    3332    3640    3641
TCHECK     116     117     122     123     304     305     769     770    2084    2085    2426    2427    2737    2738
          3035    3036    3333    3334    3642    3643
TEXEC       16#     95     285     749    1718    2044    2386    2715    3013    3311    3620
TEXIT      117     118     123     124     305     306     770     771    1739    1740    2738    2739    3036    3037
          3334    3335    3643    3644
TG40        67      79#
TG41       258     269#
TG42       494     505#
TG43       606     616#
TG44       721     733#
TG45       945     955#
TG46      1146    1155#
TG47      1316    1325#
TG47A     1331    1335#
TG50      1564    1572#
TG51      1692    1701#
TG52      1919    1927#
TG53      2016    2029#
TG54      2359    2371#
TG55      2689    2700#
TG56      2988    2998#
TG57      3287    3296#
TG60      3597    3605#
TJRST     2739    2740    3037    3038    3335    3336    3644    3645
TL40        87#    103
TL41       277#    293
TL42       513#    528
TL43       624#    639
TL44       741#    757
TL45       963#    978
TL46      1163#   1178
TL47      1343#   1358
TL51      1711#   1726
TL52      1935#   1950
TL53      2037#   2072
TL54      2379#   2414
TL55      2708#   2723
TL56      3006#   3021
TL57      3304#   3319
TL60      3613#   3628
TLAST      124     125     306     307     771     772    1740    1741    2085    2086    2427    2428    2740    2741
```

1958

```
           3038      3039      3336      3337      3645      3646
TLOAD        15#       82       272       508       619       736        958       1158       1338       1581       1704       1930       2032       2374        SEQ 1971
           2703      3001      3299      3608
TRACE        15#       80       270       506       617       734        956       1156       1336       1579       1702       1928       2030       2372
           2701      2999      3297      3606
TS40         90       115#
TS41        280       303#
TS42        516       538#
TS43        627       649#
TS44        742       767#
TS44DA      767       775#
TS45        964       988#
TS45CH      991       996#
TS46       1164      1188#
TS46CH     1191      1196#
TS47       1344      1375#
TS47CH     1379      1400#
TS47DO     1387#     1395
TS47DA     1376      1384#
TS51       1712      1736#
TS51CO     1769      1784#
TS51CD     1706      1801#
TS51CH     1738      1763#
TS51IN     1736      1744#
TS51TB     1766      1767      1773      1784      1787      1790      1798#      1822
TS51TC     1706      1707      1774      1776      1780      1785      1788       1791      1799#      1816
TS52       1938      1960#
TS53       2040      2082#
TS53BU     2107      2114#     2200      2202
TS53DA     2082      2090#
TS54       2382      2424#
TS54BU     2449      2456#     2529      2531
TS54DA     2424      2432#
TS55       2711      2733#
TS55A      2735#     2739      2781
TS55BU     2755      2786#     2803      2805
TS55DA     2733      2744#
TS55IN     2734      2772#
TS55NE     2735      2777#
TS56       3009      3031#
TS56A      3033#     3037      3079
TS56BU     3053      3084#     3101      3103
TS56DA     3031      3042#
TS56IN     3032      3070#
TS56NE     3033      3075#
TS57       3307      3329#
TS57A      3331#     3335      3377
TS57BU     3351      3382#     3401      3403
TS57DA     3329      3340#
TS57IN     3330      3368#
TS57NE     3331      3373#
TS60       3616      3638#
TS60A      3640#     3644      3695
```

```
TS60BU       3660       3700#      3721       3723
TS60DA       3638       3649#
TS60IN       3639       3677#
TS60NE       3640       3691#
TSSTAR       2736       3034       3332       3641
TSTART        115        116        121        122        303        304        768        769       1737       1738       2083       2084       2425       2426
TSTC40         10         67#
TSTC41         10         71        258#
TSTC42         10         72        262        494#
TSTC43         10         73        263        498        606#
TSTC44         10         74        264        499        610        721#
TSTC45         10        725        945#
TSTC46         10        726        949       1146#
TSTC47         10        727        950       1150       1316#
TSTC50         10       1564#
TSTC51         11       1692#
TSTC52         11        500        611        728       1320       1696       1919#
TSTC53         11       2016#
TSTC54         11       2020       2359#
TSTC55         11       2021       2363       2689#
TSTC56         11       2022       2364       2693       2988#
TSTC57         11       2023       2365       2694       2992       3287#
TSTC60         11       2024       2366       2695       2993       3291       3597#
TSTEBF        318        328        338       1037       1445       1629       2054       2064       2396       2406
TSTFLG         15#      1327       1330       1573       1576
TSTSUB         15#        89        131        279        515        626        743        965       1165       1345       1713       1937       2039       2058
             2059       2381       2400       2401       2710       3008       3306       3615
TTPNT          16#       129        311        794       2196       2525       2799       3097       3397       3717
TUSER        1328       1574
TX40           96        104        109#
TX41          286        294        299#
TX42          521        529        534#
TX43          632        640        645#
TX44          750        758        763#
TX45          971        979        984#
TX46         1171       1179       1184#
TX47         1339       1351       1359       1364#
TX50         1614       1618#
TX51         1719       1727       1732#
TX52         1943       1951       1956#
TX53         2045       2073       2078#
TX54         2387       2415       2420#
TX55         2716       2724       2729#
TX56         3014       3022       3027#
TX57         3312       3320       3325#
TX60         3621       3629       3634#
TXALL         129        311        547        658        794       1008       1208       1416       1636       1809       1971       2196       2525       2799
             3097       3397       3717
TXNOT         128        310        546        657        793       1006       1007       1206       1207       1414       1415       1808       1969       2194
             2195       2523       2524       2797       2798       3095       3096       3395       3396       3715       3716
TXTINH       1640       1812       1814       1818       1820       1828
UDEBUG         15#      1329       1575       1778
USER         1326       1388       1391       1572       1768
```

```
1960  ZA          31#     79    269    505    616    733    955   1155   1335   1578   1701   1927   2029   2371    SEQ 1973
                2700   2998   3296   3605
      ZCBUS       68    259    495    607    722    946   1147   1317   1565   1693   1920   2017   2360   2690
                2989   3288   3598
      $ARG2     142#    142   144#    144   146#    146   148#    148   150#    150   152#    152   154#    154
                156#    156   158#    158   163#    163   165#    165   167#    167   169#    169   171#    171
                173#    173   175#    175   177#    177   179#    179   184#    184   186#    186   188#    188
                190#    190   192#    192   194#    194   196#    196   198#    198   200#    200   205#    205
                207#    207   209#    209   211#    211   213#    213   215#    215   217#    217   219#    219
                221#    221   360#    360   362#    362   364#    364   366#    366   368#    368   371#    371
                376#    376   378#    378   380#    380   382#    382   387#    387   389#    389   391#    391
                393#    393   398#    398   400#    400   406#    406   408#    408   410#    410   415#    415
                417#    417   419#    419   424#    424   426#    426   428#    428   430#    430   432#    432
                434#    434   436#    436   438#    438   440#    440   445#    445   447#    447   449#    449
                451#    451   453#    453   455#    455   457#    457   459#    459   461#    461   551#    551
                553#    553   555#    555   557#    557   559#    559   562#    562   564#    564   566#    566
                568#    568   570#    570   572#    572   662#    662   664#    664   666#    666   668#    668
                670#    670   673#    673   675#    675   677#    677   679#    679   681#    681   683#    683
                800#    800   802#    802   804#    804   806#    806   808#    808   813#    813   815#    815
                817#    817   819#    819   824#    824   826#    826   828#    828   830#    830   832#    832
                837#    837   839#    839   841#    841   843#    843   848#    848   853#    853   855#    855
                857#    857   859#    859   864#    864   866#    866   869#    869   874#    874   876#    876
                878#    878   880#    880   882#    882   884#    884   886#    886   888#    888   890#    890
                895#    895   897#    897   899#    899   901#    901   903#    903   905#    905   907#    907
                909#    909   911#    911  1051#   1051  1053#   1053  1055#   1055  1057#   1057  1059#   1059
               1061#   1061  1063#   1063  1068#   1068  1070#   1070  1072#   1072  1077#   1077  1079#   1079
               1081#   1081  1088#   1088  1090#   1090  1092#   1092  1094#   1094  1096#   1096  1098#   1098
               1103#   1103  1105#   1105  1110#   1110  1115#   1115  1117#   1117  1240#   1240  1242#   1242
               1247#   1247  1249#   1249  1251#   1251  1253#   1253  1255#   1255  1257#   1257  1259#   1259
               1261#   1261  1266#   1266  1268#   1268  1270#   1270  1275#   1275  1280#   1280  1282#   1282
               1459#   1459  1461#   1461  1463#   1463  1465#   1465  1467#   1467  1469#   1469  1471#   1471
               1473#   1473  1479#   1479  1481#   1481  1483#   1483  1488#   1488  1490#   1490  1492#   1492
               1499#   1499  1501#   1501  1503#   1503  1505#   1505  1507#   1507  1509#   1509  1514#   1514
               1516#   1516  1521#   1521  1526#   1526  1528#   1528  1662#   1662  1664#   1664  1667#   1667
               1669#   1669  1879#   1879  1881#   1881  1883#   1883  1885#   1885  1887#   1887  1889#   1889
               1891#   1891  1896#   1896  1898#   1898  1975#   1975  1977#   1977  1979#   1979  1982#   1982
               1984#   1984  2213#   2213  2215#   2215  2217#   2217  2219#   2219  2221#   2221  2223#   2223
               2228#   2228  2230#   2230  2232#   2232  2234#   2234  2236#   2236  2238#   2238  2240#   2240
               2242#   2242  2247#   2247  2254#   2254  2256#   2256  2258#   2258  2260#   2260  2262#   2262
               2264#   2264  2266#   2266  2268#   2268  2273#   2273  2278#   2278  2283#   2283  2285#   2285
               2290#   2290  2292#   2292  2294#   2294  2296#   2296  2298#   2298  2300#   2300  2302#   2302
               2304#   2304  2306#   2306  2311#   2311  2313#   2313  2315#   2315  2317#   2317  2319#   2319
               2321#   2321  2323#   2323  2325#   2325  2327#   2327  2542#   2542  2544#   2544  2546#   2546
               2548#   2548  2550#   2550  2552#   2552  2557#   2557  2559#   2559  2561#   2561  2563#   2563
               2565#   2565  2567#   2567  2569#   2569  2571#   2571  2576#   2576  2583#   2583  2585#   2585
               2587#   2587  2589#   2589  2591#   2591  2593#   2593  2595#   2595  2597#   2597  2602#   2602
               2607#   2607  2612#   2612  2614#   2614  2619#   2619  2621#   2621  2623#   2623  2625#   2625
               2627#   2627  2629#   2629  2631#   2631  2633#   2633  2635#   2635  2640#   2640  2642#   2642
               2644#   2644  2646#   2646  2648#   2648  2650#   2650  2652#   2652  2654#   2654  2656#   2656
               2816#   2816  2818#   2818  2820#   2820  2822#   2822  2824#   2824  2826#   2826  2828#   2828
               2830#   2830  2835#   2835  2837#   2837  2839#   2839  2841#   2841  2843#   2843  2845#   2845
               2847#   2847  2849#   2849  2854#   2854  2856#   2856  2858#   2858  2860#   2860  2862#   2862
               2867#   2867  2872#   2872  2874#   2874  2876#   2876  2878#   2878  2880#   2880  2882#   2882
```

```
Q 1961  2884# 2884   2886# 2886   2888# 2888   2890# 2890   2892# 2892   2894# 2894   2896# 2896   SEQ 1974
        2898# 2898   2900# 2900   2902# 2902   2904# 2904   2906# 2906   2908# 2908   2910# 2910
        2915# 2915   2917# 2917   2919# 2919   2921# 2921   2923# 2923   2925# 2925   2927# 2927
        2929# 2929   2931# 2931   2936# 2936   2938# 2938   2940# 2940   2942# 2942   2944# 2944
        2946# 2946   2948# 2948   2950# 2950   2952# 2952   3114# 3114   3116# 3116   3118# 3118
        3120# 3120   3122# 3122   3124# 3124   3126# 3126   3128# 3128   3133# 3133   3135# 3135
        3137# 3137   3139# 3139   3141# 3141   3143# 3143   3145# 3145   3147# 3147   3149# 3149
        3154# 3154   3156# 3156   3158# 3158   3160# 3160   3162# 3162   3167# 3167   3172# 3172
        3174# 3174   3176# 3176   3178# 3178   3180# 3180   3182# 3182   3184# 3184   3186# 3186
        3188# 3188   3190# 3190   3192# 3192   3194# 3194   3196# 3196   3198# 3198   3200# 3200
        3202# 3202   3204# 3204   3206# 3206   3208# 3208   3210# 3210   3215# 3215   3217# 3217
        3219# 3219   3221# 3221   3223# 3223   3225# 3225   3227# 3227   3229# 3229   3231# 3231
        3236# 3236   3238# 3238   3240# 3240   3242# 3242   3244# 3244   3246# 3246   3248# 3248
        3250# 3250   3252# 3252   3414# 3414   3416# 3416   3418# 3418   3420# 3420   3422# 3422
        3424# 3424   3426# 3426   3428# 3428   3430# 3430   3432# 3432   3437# 3437   3439# 3439
        3441# 3441   3443# 3443   3445# 3445   3447# 3447   3449# 3449   3451# 3451   3453# 3453
        3458# 3458   3460# 3460   3462# 3462   3464# 3464   3466# 3466   3471# 3471   3476# 3476
        3478# 3478   3480# 3480   3482# 3482   3484# 3484   3486# 3486   3488# 3488   3490# 3490
        3492# 3492   3494# 3494   3496# 3496   3498# 3498   3500# 3500   3502# 3502   3504# 3504
        3506# 3506   3508# 3508   3510# 3510   3512# 3512   3514# 3514   3516# 3516   3518# 3518
        3520# 3520   3522# 3522   3527# 3527   3529# 3529   3531# 3531   3533# 3533   3535# 3535
        3537# 3537   3539# 3539   3541# 3541   3543# 3543   3548# 3548   3550# 3550   3552# 3552
        3554# 3554   3556# 3556   3558# 3558   3560# 3560   3562# 3562   3564# 3564   3734# 3734
        3736# 3736   3738# 3738   3740# 3740   3742# 3742   3744# 3744   3746# 3746   3748# 3748
        3750# 3750   3752# 3752   3754# 3754   3756# 3756   3761# 3761   3763# 3763   3765# 3765
        3767# 3767   3769# 3769   3771# 3771   3773# 3773   3775# 3775   3780# 3780   3782# 3782
        3784# 3784   3786# 3786   3788# 3788   3793# 3793   3798# 3798   3800# 3800   3802# 3802
        3804# 3804   3806# 3806   3808# 3808   3810# 3810   3812# 3812   3814# 3814   3816# 3816
        3818# 3818   3820# 3820   3822# 3822   3824# 3824   3826# 3826   3828# 3828   3830# 3830
        3832# 3832   3834# 3834   3836# 3836   3838# 3838   3840# 3840   3842# 3842   3844# 3844
        3846# 3846   3848# 3848   3850# 3850   3852# 3852   3857# 3857   3859# 3859   3861# 3861
        3863# 3863   3865# 3865   3867# 3867   3869# 3869   3871# 3871   3873# 3873   3878# 3878
        3880# 3880   3882# 3882   3884# 3884   3886# 3886   3888# 3888   3890# 3890   3892# 3892
        3894# 3894
$B       142#  142    144#  144    146#  146    148#  148    150#  150    152#  152    154#  154
         156#  156    158#  158    163#  163    165#  165    167#  167    169#  169    171#  171
         173#  173    175#  175    177#  177    179#  179    184#  184    186#  186    188#  188
         190#  190    192#  192    194#  194    196#  196    198#  198    200#  200    205#  205
         207#  207    209#  209    211#  211    213#  213    215#  215    217#  217    219#  219
         221#  221    360#  360    362#  362    364#  364    366#  366    368#  368    371#  371
         376#  376    378#  378    380#  380    382#  382    387#  387    389#  389    391#  391
         393#  393    398#  398    400#  400    406#  406    408#  408    410#  410    415#  415
         417#  417    419#  419    424#  424    426#  426    428#  428    430#  430    432#  432
         434#  434    436#  436    438#  438    440#  440    445#  445    447#  447    449#  449
         451#  451    453#  453    455#  455    457#  457    459#  459    461#  461    551#  551
         553#  553    555#  555    557#  557    559#  559    562#  562    564#  564    566#  566
         568#  568    570#  570    572#  572    662#  662    664#  664    666#  666    668#  668
         670#  670    673#  673    675#  675    677#  677    679#  679    681#  681    683#  683
         800#  800    802#  802    804#  804    806#  806    808#  808    813#  813    815#  815
         817#  817    819#  819    824#  824    826#  826    828#  828    830#  830    832#  832
         837#  837    839#  839    841#  841    843#  843    848#  848    853#  853    855#  855
         857#  857    859#  859    864#  864    866#  866    869#  869    874#  874    876#  876
         878#  878    880#  880    882#  882    884#  884    886#  886    888#  888    890#  890
```

```
 895#  895   897#  897   899#  899   901#  901   903#  903   905#  905   907#  907
 909#  909   911#  911  1051# 1051  1053# 1053  1055# 1055  1057# 1057  1059# 1059
1061# 1061  1063# 1063  1068# 1068  1070# 1070  1072# 1072  1077# 1077  1079# 1079
1081# 1081  1088# 1088  1090# 1090  1092# 1092  1094# 1094  1096# 1096  1098# 1098
1103# 1103  1105# 1105  1110# 1110  1115# 1115  1117# 1117  1240# 1240  1242# 1242
1247# 1247  1249# 1249  1251# 1251  1253# 1253  1255# 1255  1257# 1257  1259# 1259
1261# 1261  1266# 1266  1268# 1268  1270# 1270  1275# 1275  1280# 1280  1282# 1282
1459# 1459  1461# 1461  1463# 1463  1465# 1465  1467# 1467  1469# 1469  1471# 1471
1473# 1473  1479# 1479  1481# 1481  1483# 1483  1488# 1488  1490# 1490  1492# 1492
1499# 1499  1501# 1501  1503# 1503  1505# 1505  1507# 1507  1509# 1509  1514# 1514
1516# 1516  1521# 1521  1526# 1526  1528# 1528  1662# 1662  1664# 1664  1667# 1667
1669# 1669  1879# 1879  1881# 1881  1883# 1883  1885# 1885  1887# 1887  1889# 1889
1891# 1891  1896# 1896  1898# 1898  1975# 1975  1977# 1977  1979# 1979  1982# 1982
1984# 1984  2213# 2213  2215# 2215  2217# 2217  2219# 2219  2221# 2221  2223# 2223
2228# 2228  2230# 2230  2232# 2232  2234# 2234  2236# 2236  2238# 2238  2240# 2240
2242# 2242  2247# 2247  2254# 2254  2256# 2256  2258# 2258  2260# 2260  2262# 2262
2264# 2264  2266# 2266  2268# 2268  2273# 2273  2278# 2278  2283# 2283  2285# 2285
2290# 2290  2292# 2292  2294# 2294  2296# 2296  2298# 2298  2300# 2300  2302# 2302
2304# 2304  2306# 2306  2311# 2311  2313# 2313  2315# 2315  2317# 2317  2319# 2319
2321# 2321  2323# 2323  2325# 2325  2327# 2327  2542# 2542  2544# 2544  2546# 2546
2548# 2548  2550# 2550  2552# 2552  2557# 2557  2559# 2559  2561# 2561  2563# 2563
2565# 2565  2567# 2567  2569# 2569  2571# 2571  2576# 2576  2583# 2583  2585# 2585
2587# 2587  2589# 2589  2591# 2591  2593# 2593  2595# 2595  2597# 2597  2602# 2602
2607# 2607  2612# 2612  2614# 2614  2619# 2619  2621# 2621  2623# 2623  2625# 2625
2627# 2627  2629# 2629  2631# 2631  2633# 2633  2635# 2635  2640# 2640  2642# 2642
2644# 2644  2646# 2646  2648# 2648  2650# 2650  2652# 2652  2654# 2654  2656# 2656
2816# 2816  2818# 2818  2820# 2820  2822# 2822  2824# 2824  2826# 2826  2828# 2828
2830# 2830  2835# 2835  2837# 2837  2839# 2839  2841# 2841  2843# 2843  2845# 2845
2847# 2847  2849# 2849  2854# 2854  2856# 2856  2858# 2858  2860# 2860  2862# 2862
2867# 2867  2872# 2872  2874# 2874  2876# 2876  2878# 2878  2880# 2880  2882# 2882
2884# 2884  2886# 2886  2888# 2888  2890# 2890  2892# 2892  2894# 2894  2896# 2896
2898# 2898  2900# 2900  2902# 2902  2904# 2904  2906# 2906  2908# 2908  2910# 2910
2915# 2915  2917# 2917  2919# 2919  2921# 2921  2923# 2923  2925# 2925  2927# 2927
2929# 2929  2931# 2931  2936# 2936  2938# 2938  2940# 2940  2942# 2942  2944# 2944
2946# 2946  2948# 2948  2950# 2950  2952# 2952  3114# 3114  3116# 3116  3118# 3118
3120# 3120  3122# 3122  3124# 3124  3126# 3126  3128# 3128  3133# 3133  3135# 3135
3137# 3137  3139# 3139  3141# 3141  3143# 3143  3145# 3145  3147# 3147  3149# 3149
3154# 3154  3156# 3156  3158# 3158  3160# 3160  3162# 3162  3167# 3167  3172# 3172
3174# 3174  3176# 3176  3178# 3178  3180# 3180  3182# 3182  3184# 3184  3186# 3186
3188# 3188  3190# 3190  3192# 3192  3194# 3194  3196# 3196  3198# 3198  3200# 3200
3202# 3202  3204# 3204  3206# 3206  3208# 3208  3210# 3210  3215# 3215  3217# 3217
3219# 3219  3221# 3221  3223# 3223  3225# 3225  3227# 3227  3229# 3229  3231# 3231
3236# 3236  3238# 3238  3240# 3240  3242# 3242  3244# 3244  3246# 3246  3248# 3248
3250# 3250  3252# 3252  3414# 3414  3416# 3416  3418# 3418  3420# 3420  3422# 3422
3424# 3424  3426# 3426  3428# 3428  3430# 3430  3432# 3432  3437# 3437  3439# 3439
3441# 3441  3443# 3443  3445# 3445  3447# 3447  3449# 3449  3451# 3451  3453# 3453
3458# 3458  3460# 3460  3462# 3462  3464# 3464  3466# 3466  3471# 3471  3476# 3476
3478# 3478  3480# 3480  3482# 3482  3484# 3484  3486# 3486  3488# 3488  3490# 3490
3492# 3492  3494# 3494  3496# 3496  3498# 3498  3500# 3500  3502# 3502  3504# 3504
3506# 3506  3508# 3508  3510# 3510  3512# 3512  3514# 3514  3516# 3516  3518# 3518
3520# 3520  3522# 3522  3527# 3527  3529# 3529  3531# 3531  3533# 3533  3535# 3535
3537# 3537  3539# 3539  3541# 3541  3543# 3543  3548# 3548  3550# 3550  3552# 3552
3554# 3554  3556# 3556  3558# 3558  3560# 3560  3562# 3562  3564# 3564  3734# 3734
```

```
EQ 1963   3736#  3736    3738#  3738    3740#  3740    3742#  3742    3744#  3744    3746#  3746    3748#  3748   SEQ 1976
          3750#  3750    3752#  3752    3754#  3754    3756#  3756    3761#  3761    3763#  3763    3765#  3765
          3767#  3767    3769#  3769    3771#  3771    3773#  3773    3775#  3775    3780#  3780    3782#  3782
          3784#  3784    3786#  3786    3788#  3788    3793#  3793    3798#  3798    3800#  3800    3802#  3802
          3804#  3804    3806#  3806    3808#  3808    3810#  3810    3812#  3812    3814#  3814    3816#  3816
          3818#  3818    3820#  3820    3822#  3822    3824#  3824    3826#  3826    3828#  3828    3830#  3830
          3832#  3832    3834#  3834    3836#  3836    3838#  3838    3840#  3840    3842#  3842    3844#  3844
          3846#  3846    3848#  3848    3850#  3850    3852#  3852    3857#  3857    3859#  3859    3861#  3861
          3863#  3863    3865#  3865    3867#  3867    3869#  3869    3871#  3871    3873#  3873    3878#  3878
          3880#  3880    3882#  3882    3884#  3884    3886#  3886    3888#  3888    3890#  3890    3892#  3892
          3894#  3894
$CHR       142#   142     144#   144     146#   146     148#   148     150#   150     152#   152     154#   154
           156#   156     158#   158     163#   163     165#   165     167#   167     169#   169     171#   171
           173#   173     175#   175     177#   177     179#   179     184#   184     186#   186     188#   188
           190#   190     192#   192     194#   194     196#   196     198#   198     200#   200     205#   205
           207#   207     209#   209     211#   211     213#   213     215#   215     217#   217     219#   219
           221#   221     360#   360     362#   362     364#   364     366#   366     368#   368     371#   371
           376#   376     378#   378     380#   380     382#   382     387#   387     389#   389     391#   391
           393#   393     398#   398     400#   400     406#   406     408#   408     410#   410     415#   415
           417#   417     419#   419     424#   424     426#   426     428#   428     430#   430     432#   432
           434#   434     436#   436     438#   438     440#   440     445#   445     447#   447     449#   449
           451#   451     453#   453     455#   455     457#   457     459#   459     461#   461     551#   551
           553#   553     555#   555     557#   557     559#   559     562#   562     564#   564     566#   566
           568#   568     570#   570     572#   572     662#   662     664#   664     666#   666     668#   668
           670#   670     673#   673     675#   675     677#   677     679#   679     681#   681     683#   683
           800#   800     802#   802     804#   804     806#   806     808#   808     813#   813     815#   815
           817#   817     819#   819     824#   824     826#   826     828#   828     830#   830     832#   832
           837#   837     839#   839     841#   841     843#   843     848#   848     853#   853     855#   855
           857#   857     859#   859     864#   864     866#   866     869#   869     874#   874     876#   876
           878#   878     880#   880     882#   882     884#   884     886#   886     888#   888     890#   890
           895#   895     897#   897     899#   899     901#   901     903#   903     905#   905     907#   907
           909#   909     911#   911    1051#  1051    1053#  1053    1055#  1055    1057#  1057    1059#  1059
          1061#  1061    1063#  1063    1068#  1068    1070#  1070    1072#  1072    1077#  1077    1079#  1079
          1081#  1081    1088#  1088    1090#  1090    1092#  1092    1094#  1094    1096#  1096    1098#  1098
          1103#  1103    1105#  1105    1110#  1110    1115#  1115    1117#  1117    1240#  1240    1242#  1242
          1247#  1247    1249#  1249    1251#  1251    1253#  1253    1255#  1255    1257#  1257    1259#  1259
          1261#  1261    1266#  1266    1268#  1268    1270#  1270    1275#  1275    1280#  1280    1282#  1282
          1459#  1459    1461#  1461    1463#  1463    1465#  1465    1467#  1467    1469#  1469    1471#  1471
          1473#  1473    1479#  1479    1481#  1481    1483#  1483    1488#  1488    1490#  1490    1492#  1492
          1499#  1499    1501#  1501    1503#  1503    1505#  1505    1507#  1507    1509#  1509    1514#  1514
          1516#  1516    1521#  1521    1526#  1526    1528#  1528    1662#  1662    1664#  1664    1667#  1667
          1669#  1669    1879#  1879    1881#  1881    1883#  1883    1885#  1885    1887#  1887    1889#  1889
          1891#  1891    1896#  1896    1898#  1898    1975#  1975    1977#  1977    1979#  1979    1982#  1982
          1984#  1984    2213#  2213    2215#  2215    2217#  2217    2219#  2219    2221#  2221    2223#  2223
          2228#  2228    2230#  2230    2232#  2232    2234#  2234    2236#  2236    2238#  2238    2240#  2240
          2242#  2242    2247#  2247    2254#  2254    2256#  2256    2258#  2258    2260#  2260    2262#  2262
          2264#  2264    2266#  2266    2268#  2268    2273#  2273    2278#  2278    2283#  2283    2285#  2285
          2290#  2290    2292#  2292    2294#  2294    2296#  2296    2298#  2298    2300#  2300    2302#  2302
          2304#  2304    2306#  2306    2311#  2311    2313#  2313    2315#  2315    2317#  2317    2319#  2319
          2321#  2321    2323#  2323    2325#  2325    2327#  2327    2542#  2542    2544#  2544    2546#  2546
          2548#  2548    2550#  2550    2552#  2552    2557#  2557    2559#  2559    2561#  2561    2563#  2563
          2565#  2565    2567#  2567    2569#  2569    2571#  2571    2576#  2576    2583#  2583    2585#  2585
          2587#  2587    2589#  2589    2591#  2591    2593#  2593    2595#  2595    2597#  2597    2602#  2602
```

```
               2607#  2607   2612#  2612   2614#  2614   2619#  2619   2621#  2621   2623#  2623   2625#  2625      SEQ 1977
               2627#  2627   2629#  2629   2631#  2631   2633#  2633   2635#  2635   2640#  2640   2642#  2642
               2644#  2644   2646#  2646   2648#  2648   2650#  2650   2652#  2652   2654#  2654   2656#  2656
               2816#  2816   2818#  2818   2820#  2820   2822#  2822   2824#  2824   2826#  2826   2828#  2828
               2830#  2830   2835#  2835   2837#  2837   2839#  2839   2841#  2841   2843#  2843   2845#  2845
               2847#  2847   2849#  2849   2854#  2854   2856#  2856   2858#  2858   2860#  2860   2862#  2862
               2867#  2867   2872#  2872   2874#  2874   2876#  2876   2878#  2878   2880#  2880   2882#  2882
               2884#  2884   2886#  2886   2888#  2888   2890#  2890   2892#  2892   2894#  2894   2896#  2896
               2898#  2898   2900#  2900   2902#  2902   2904#  2904   2906#  2906   2908#  2908   2910#  2910
               2915#  2915   2917#  2917   2919#  2919   2921#  2921   2923#  2923   2925#  2925   2927#  2927
               2929#  2929   2931#  2931   2936#  2936   2938#  2938   2940#  2940   2942#  2942   2944#  2944
               2946#  2946   2948#  2948   2950#  2950   2952#  2952   3114#  3114   3116#  3116   3118#  3118
               3120#  3120   3122#  3122   3124#  3124   3126#  3126   3128#  3128   3133#  3133   3135#  3135
               3137#  3137   3139#  3139   3141#  3141   3143#  3143   3145#  3145   3147#  3147   3149#  3149
               3154#  3154   3156#  3156   3158#  3158   3160#  3160   3162#  3162   3167#  3167   3172#  3172
               3174#  3174   3176#  3176   3178#  3178   3180#  3180   3182#  3182   3184#  3184   3186#  3186
               3188#  3188   3190#  3190   3192#  3192   3194#  3194   3196#  3196   3198#  3198   3200#  3200
               3202#  3202   3204#  3204   3206#  3206   3208#  3208   3210#  3210   3215#  3215   3217#  3217
               3219#  3219   3221#  3221   3223#  3223   3225#  3225   3227#  3227   3229#  3229   3231#  3231
               3236#  3236   3238#  3238   3240#  3240   3242#  3242   3244#  3244   3246#  3246   3248#  3248
               3250#  3250   3252#  3252   3414#  3414   3416#  3416   3418#  3418   3420#  3420   3422#  3422
               3424#  3424   3426#  3426   3428#  3428   3430#  3430   3432#  3432   3437#  3437   3439#  3439
               3441#  3441   3443#  3443   3445#  3445   3447#  3447   3449#  3449   3451#  3451   3453#  3453
               3458#  3458   3460#  3460   3462#  3462   3464#  3464   3466#  3466   3471#  3471   3476#  3476
               3478#  3478   3480#  3480   3482#  3482   3484#  3484   3486#  3486   3488#  3488   3490#  3490
               3492#  3492   3494#  3494   3496#  3496   3498#  3498   3500#  3500   3502#  3502   3504#  3504
               3506#  3506   3508#  3508   3510#  3510   3512#  3512   3514#  3514   3516#  3516   3518#  3518
               3520#  3520   3522#  3522   3527#  3527   3529#  3529   3531#  3531   3533#  3533   3535#  3535
               3537#  3537   3539#  3539   3541#  3541   3543#  3543   3548#  3548   3550#  3550   3552#  3552
               3554#  3554   3556#  3556   3558#  3558   3560#  3560   3562#  3562   3564#  3564   3734#  3734
               3736#  3736   3738#  3738   3740#  3740   3742#  3742   3744#  3744   3746#  3746   3748#  3748
               3750#  3750   3752#  3752   3754#  3754   3756#  3756   3761#  3761   3763#  3763   3765#  3765
               3767#  3767   3769#  3769   3771#  3771   3773#  3773   3775#  3775   3780#  3780   3782#  3782
               3784#  3784   3786#  3786   3788#  3788   3793#  3793   3798#  3798   3800#  3800   3802#  3802
               3804#  3804   3806#  3806   3808#  3808   3810#  3810   3812#  3812   3814#  3814   3816#  3816
               3818#  3818   3820#  3820   3822#  3822   3824#  3824   3826#  3826   3828#  3828   3830#  3830
               3832#  3832   3834#  3834   3836#  3836   3838#  3838   3840#  3840   3842#  3842   3844#  3844
               3846#  3846   3848#  3848   3850#  3850   3852#  3852   3857#  3857   3859#  3859   3861#  3861
               3863#  3863   3865#  3865   3867#  3867   3869#  3869   3871#  3871   3873#  3873   3878#  3878
               3880#  3880   3882#  3882   3884#  3884   3886#  3886   3888#  3888   3890#  3890   3892#  3892
               3894#  3894
$GARG           142#   142    144#   144    146#   146    148#   148    150#   150    152#   152    154#   154
                156#   156    158#   158    163#   163    165#   165    167#   167    169#   169    171#   171
                173#   173    175#   175    177#   177    179#   179    184#   184    186#   186    188#   188
                190#   190    192#   192    194#   194    196#   196    198#   198    200#   200    205#   205
                207#   207    209#   209    211#   211    213#   213    215#   215    217#   217    219#   219
                221#   221    360#   360    362#   362    364#   364    366#   366    368#   368    371#   371
                376#   376    378#   378    380#   380    382#   382    387#   387    389#   389    391#   391
                393#   393    398#   398    400#   400    406#   406    408#   408    410#   410    415#   415
                417#   417    419#   419    424#   424    426#   426    428#   428    430#   430    432#   432
                434#   434    436#   436    438#   438    440#   440    445#   445    447#   447    449#   449
                451#   451    453#   453    455#   455    457#   457    459#   459    461#   461    551#   551
                553#   553    555#   555    557#   557    559#   559    562#   562    564#   564    566#   566
```

```
 568#  568   570#  570   572#  572   662#  662   664#  664   666#  666   668#  668      SEQ 1978
 670#  670   673#  673   675#  675   677#  677   679#  679   681#  681   683#  683
 800#  800   802#  802   804#  804   806#  806   808#  808   813#  813   815#  815
 817#  817   819#  819   824#  824   826#  826   828#  828   830#  830   832#  832
 837#  837   839#  839   841#  841   843#  843   848#  848   853#  853   855#  855
 857#  857   859#  859   864#  864   866#  866   869#  869   874#  874   876#  876
 878#  878   880#  880   882#  882   884#  884   886#  886   888#  888   890#  890
 895#  895   897#  897   899#  899   901#  901   903#  903   905#  905   907#  907
 909#  909   911#  911  1051# 1051  1053# 1053  1055# 1055  1057# 1057  1059# 1059
1061# 1061  1063# 1063  1068# 1068  1070# 1070  1072# 1072  1077# 1077  1079# 1079
1081# 1081  1088# 1088  1090# 1090  1092# 1092  1094# 1094  1096# 1096  1098# 1098
1103# 1103  1105# 1105  1110# 1110  1115# 1115  1117# 1117  1240# 1240  1242# 1242
1247# 1247  1249# 1249  1251# 1251  1253# 1253  1255# 1255  1257# 1257  1259# 1259
1261# 1261  1266# 1266  1268# 1268  1270# 1270  1275# 1275  1280# 1280  1282# 1282
1459# 1459  1461# 1461  1463# 1463  1465# 1465  1467# 1467  1469# 1469  1471# 1471
1473# 1473  1479# 1479  1481# 1481  1483# 1483  1488# 1488  1490# 1490  1492# 1492
1499# 1499  1501# 1501  1503# 1503  1505# 1505  1507# 1507  1509# 1509  1514# 1514
1516# 1516  1521# 1521  1526# 1526  1528# 1528  1662# 1662  1664# 1664  1667# 1667
1669# 1669  1879# 1879  1881# 1881  1883# 1883  1885# 1885  1887# 1887  1889# 1889
1891# 1891  1896# 1896  1898# 1898  1975# 1975  1977# 1977  1979# 1979  1982# 1982
1984# 1984  2213# 2213  2215# 2215  2217# 2217  2219# 2219  2221# 2221  2223# 2223
2228# 2228  2230# 2230  2232# 2232  2234# 2234  2236# 2236  2238# 2238  2240# 2240
2242# 2242  2247# 2247  2254# 2254  2256# 2256  2258# 2258  2260# 2260  2262# 2262
2264# 2264  2266# 2266  2268# 2268  2273# 2273  2278# 2278  2283# 2283  2285# 2285
2290# 2290  2292# 2292  2294# 2294  2296# 2296  2298# 2298  2300# 2300  2302# 2302
2304# 2304  2306# 2306  2311# 2311  2313# 2313  2315# 2315  2317# 2317  2319# 2319
2321# 2321  2323# 2323  2325# 2325  2327# 2327  2542# 2542  2544# 2544  2546# 2546
2548# 2548  2550# 2550  2552# 2552  2557# 2557  2559# 2559  2561# 2561  2563# 2563
2565# 2565  2567# 2567  2569# 2569  2571# 2571  2576# 2576  2583# 2583  2585# 2585
2587# 2587  2589# 2589  2591# 2591  2593# 2593  2595# 2595  2597# 2597  2602# 2602
2607# 2607  2612# 2612  2614# 2614  2619# 2619  2621# 2621  2623# 2623  2625# 2625
2627# 2627  2629# 2629  2631# 2631  2633# 2633  2635# 2635  2640# 2640  2642# 2642
2644# 2644  2646# 2646  2648# 2648  2650# 2650  2652# 2652  2654# 2654  2656# 2656
2816# 2816  2818# 2818  2820# 2820  2822# 2822  2824# 2824  2826# 2826  2828# 2828
2830# 2830  2835# 2835  2837# 2837  2839# 2839  2841# 2841  2843# 2843  2845# 2845
2847# 2847  2849# 2849  2854# 2854  2856# 2856  2858# 2858  2860# 2860  2862# 2862
2867# 2867  2872# 2872  2874# 2874  2876# 2876  2878# 2878  2880# 2880  2882# 2882
2884# 2884  2886# 2886  2888# 2888  2890# 2890  2892# 2892  2894# 2894  2896# 2896
2898# 2898  2900# 2900  2902# 2902  2904# 2904  2906# 2906  2908# 2908  2910# 2910
2915# 2915  2917# 2917  2919# 2919  2921# 2921  2923# 2923  2925# 2925  2927# 2927
2929# 2929  2931# 2931  2936# 2936  2938# 2938  2940# 2940  2942# 2942  2944# 2944
2946# 2946  2948# 2948  2950# 2950  2952# 2952  3114# 3114  3116# 3116  3118# 3118
3120# 3120  3122# 3122  3124# 3124  3126# 3126  3128# 3128  3133# 3133  3135# 3135
3137# 3137  3139# 3139  3141# 3141  3143# 3143  3145# 3145  3147# 3147  3149# 3149
3154# 3154  3156# 3156  3158# 3158  3160# 3160  3162# 3162  3167# 3167  3172# 3172
3174# 3174  3176# 3176  3178# 3178  3180# 3180  3182# 3182  3184# 3184  3186# 3186
3188# 3188  3190# 3190  3192# 3192  3194# 3194  3196# 3196  3198# 3198  3200# 3200
3202# 3202  3204# 3204  3206# 3206  3208# 3208  3210# 3210  3215# 3215  3217# 3217
3219# 3219  3221# 3221  3223# 3223  3225# 3225  3227# 3227  3229# 3229  3231# 3231
3236# 3236  3238# 3238  3240# 3240  3242# 3242  3244# 3244  3246# 3246  3248# 3248
3250# 3250  3252# 3252  3414# 3414  3416# 3416  3418# 3418  3420# 3420  3422# 3422
3424# 3424  3426# 3426  3428# 3428  3430# 3430  3432# 3432  3437# 3437  3439# 3439
3441# 3441  3443# 3443  3445# 3445  3447# 3447  3449# 3449  3451# 3451  3453# 3453
```

```
        3458#   3458    3460#   3460    3462#   3462    3464#   3464    3466#   3466    3471#   3471    3476#   3476       SEQ 1979
        3478#   3478    3480#   3480    3482#   3482    3484#   3484    3486#   3486    3488#   3488    3490#   3490
        3492#   3492    3494#   3494    3496#   3496    3498#   3498    3500#   3500    3502#   3502    3504#   3504
        3506#   3506    3508#   3508    3510#   3510    3512#   3512    3514#   3514    3516#   3516    3518#   3518
        3520#   3520    3522#   3522    3527#   3527    3529#   3529    3531#   3531    3533#   3533    3535#   3535
        3537#   3537    3539#   3539    3541#   3541    3543#   3543    3548#   3548    3550#   3550    3552#   3552
        3554#   3554    3556#   3556    3558#   3558    3560#   3560    3562#   3562    3564#   3564    3734#   3734
        3736#   3736    3738#   3738    3740#   3740    3742#   3742    3744#   3744    3746#   3746    3748#   3748
        3750#   3750    3752#   3752    3754#   3754    3756#   3756    3761#   3761    3763#   3763    3765#   3765
        3767#   3767    3769#   3769    3771#   3771    3773#   3773    3775#   3775    3780#   3780    3782#   3782
        3784#   3784    3786#   3786    3788#   3788    3793#   3793    3798#   3798    3800#   3800    3802#   3802
        3804#   3804    3806#   3806    3808#   3808    3810#   3810    3812#   3812    3814#   3814    3816#   3816
        3818#   3818    3820#   3820    3822#   3822    3824#   3824    3826#   3826    3828#   3828    3830#   3830
        3832#   3832    3834#   3834    3836#   3836    3838#   3838    3840#   3840    3842#   3842    3844#   3844
        3846#   3846    3848#   3848    3650#   3850    3852#   3852    3857#   3857    3859#   3859    3861#   3861
        3863#   3863    3865#   3865    3867#   3867    3869#   3869    3871#   3871    3873#   3873    3878#   3878
        3880#   3880    3882#   3882    3884#   3884    3886#   3886    3888#   3888    3890#   3890    3892#   3892
        3894#   3894
%ADDR   142#    142     144     144#    146     146#    148     148#    150     150#    152     152#    154     154#
        156     156#    158     158#    163#    163     165     165#    167     167#    169     169#    171     171#
        173     173#    175     175#    177     177#    179     179#    184#    184     186     186#    188     188#
        190     190#    192     192#    194     194#    196     196#    198     198#    200     200#    205#    205
        207     207#    209     209#    211     211#    213     213#    215     215#    217     217#    219     219#
        221     221#    360#    360     362     362#    364     364#    366     366#    368     368#    371     371#
        376     376#    378     378#    380     380#    382     382#    387     387#    389     389#    391     391#
        393     393#    398     398#    400     400#    406#    406     408     408#    410     410#    415#    415
        417     417#    419     419#    424#    424     426     426#    428     428#    430     430#    432     432#
        434     434#    436     436#    438     438#    440     440#    445#    445     447     447#    449     449#
        451     451#    453     453#    455     455#    457     457#    459     459#    461     461#    551#    551
        553     553#    555     555#    557     557#    559     559#    562#    562     564     564#    566     566#
        568     568#    570     570#    572     572#    662#    662     664     664#    666     666#    668     668#
        670     670#    673#    673     675     675#    677     677#    679     679#    681     681#    683     683#
        800#    800     802     802#    804     804#    806     806#    808     808#    813     813#    815     815#
        817     817#    819     819#    824     824#    826     826#    828     828#    830     830#    832     832#
        837     837#    839     839#    841     841#    843     843#    848     848#    853#    853     855     855#
        857     857#    859     859#    864#    864     866     866#    869     869#    874#    874     876     876#
        878     878#    880     880#    882     882#    884     884#    886     886#    888     888#    890     890#
        895#    895     897     897#    899     899#    901     901#    903     903#    905     905#    907     907#
        909     909#    911     911#    1051#   1051    1053    1053#   1055    1055#   1057    1057#   1059    1059#
        1061    1061#   1063    1063#   1068    1068#   1070    1070#   1072    1072#   1077    1077#   1079    1079#
        1081    1081#   1088#   1088    1090    1090#   1092    1092#   1094    1094#   1096    1096#   1098    1098#
        1103#   1103    1105    1105#   1110#   1110    1115#   1115    1117    1117#   1240#   1240    1242    1242#
        1247    1247#   1249    1249#   1251    1251#   1253    1253#   1255    1255#   1257    1257#   1259    1259#
        1261    1261#   1266    1266#   1268    1268#   1270    1270#   1275#   1275    1280#   1280    1282    1282#
        1459#   1459    1461    1461#   1463    1463#   1465    1465#   1467    1467#   1469    1469#   1471    1471#
        1473    1473#   1479    1479#   1481    1481#   1483    1483#   1488    1488#   1490    1490#   1492    1492#
        1499#   1499    1501    1501#   1503    1503#   1505    1505#   1507    1507#   1509    1509#   1514#   1514
        1516    1516#   1521#   1521    1526#   1526    1528    1528#   1662#   1662    1664    1664#   1667#   1667
        1669    1669#   1879#   1879    1881    1881#   1883    1883#   1885    1885#   1887    1887#   1889    1889#
        1891    1891#   1896#   1896    1898    1898#   1975#   1975    1977    1977#   1979    1979#   1982    1982#
        1984    1984#   2213#   2213    2215#   2215    2217    2217#   2219    2219#   2221    2221#   2223    2223#
        2228    2228#   2230    2230#   2232    2232#   2234    2234#   2236    2236#   2238    2238#   2240    2240#
        2242    2242#   2247    2247#   2254#   2254    2256    2256#   2258    2258#   2260    2260#   2262    2262#
```

```
967
          2264    2264#   2266    2266#   2268    2268#   2273#   2273    2278#   2278    2283#   2283    2285    2285#      SEQ 1980
          2290#   2290    2292    2292#   2294    2294#   2296    2296#   2298    2298#   2300    2300#   2302    2302#
          2304    2304#   2306    2306#   2311#   2311    2313    2313#   2315    2315#   2317    2317#   2319    2319#
          2321    2321#   2323    2323#   2325    2325#   2327    2327#   2542#   2542    2544#   2544    2546    2546#
          2548    2548#   2550    2550#   2552    2552#   2557    2557#   2559    2559#   2561    2561#   2563    2563#
          2565    2565#   2567    2567#   2569    2569#   2571    2571#   2576    2576#   2583#   2583    2585    2585#
          2587    2587#   2589    2589#   2591    2591#   2593    2593#   2595    2595#   2597    2597#   2602#   2602
          2607#   2607    2612#   2612    2614    2614#   2619#   2619    2621    2621#   2623    2623#   2625    2625#
          2627    2627#   2629    2629#   2631    2631#   2633    2633#   2635    2635#   2640#   2640    2642    2642#
          2644    2644#   2646    2646#   2648    2648#   2650    2650#   2652    2652#   2654    2654#   2656    2656#
          2816#   2816    2818    2818#   2820    2820#   2822    2822#   2824    2824#   2826    2826#   2828    2828#
          2830    2830#   2835#   2835    2837    2837#   2839    2839#   2841    2841#   2843    2843#   2845    2845#
          2847    2847#   2849    2849#   2854#   2854    2856    2856#   2858    2858#   2860    2860#   2862    2862#
          2867#   2867    2872#   2872    2874    2874#   2876    2876#   2878    2878#   2880    2880#   2882    2882#
          2884    2884#   2886    2886#   2888    2888#   2890    2890#   2892    2892#   2894    2894#   2896    2896#
          2898    2898#   2900    2900#   2902    2902#   2904    2904#   2906    2906#   2908    2908#   2910#   2910
          2915#   2915    2917    2917#   2919    2919#   2921    2921#   2923    2923#   2925    2925#   2927    2927#
          2929    2929#   2931    2931#   2936#   2936    2938    2938#   2940    2940#   2942    2942#   2944    2944#
          2946    2946#   2948    2948#   2950    2950#   2952    2952#   3114#   3114    3116    3116#   3118    3118#
          3120    3120#   3122    3122#   3124    3124#   3126    3126#   3128    3128#   3133#   3133    3135    3135#
          3137    3137#   3139    3139#   3141    3141#   3143    3143#   3145    3145#   3147    3147#   3149    3149#
          3154#   3154    3156    3156#   3158    3158#   3160    3160#   3162    3162#   3167#   3167    3172#   3172
          3174    3174#   3176    3176#   3178    3178#   3180    3180#   3182    3182#   3184    3184#   3186    3186#
          3188    3188#   3190    3190#   3192    3192#   3194    3194#   3196    3196#   3198    3198#   3200    3200#
          3202    3202#   3204    3204#   3206    3206#   3208    3208#   3210#   3210    3215#   3215    3217    3217#
          3219    3219#   3221    3221#   3223    3223#   3225    3225#   3227    3227#   3229    3229#   3231    3231#
          3236#   3236    3238    3238#   3240    3240#   3242    3242#   3244    3244#   3246    3246#   3248    3248#
          3250    3250#   3252    3252#   3414#   3414    3416    3416#   3418    3418#   3420    3420#   3422    3422#
          3424    3424#   3426    3426#   3428    3428#   3430    3430#   3432    3432#   3437#   3437    3439    3439#
          3441    3441#   3443    3443#   3445    3445#   3447    3447#   3449    3449#   3451    3451#   3453    3453#
          3458#   3458    3460    3460#   3462    3462#   3464    3464#   3466    3466#   3471#   3471    3476#   3476
          3478    3478#   3480    3480#   3482    3482#   3484    3484#   3486    3486#   3488    3488#   3490    3490#
          3492    3492#   3494    3494#   3496    3496#   3498    3498#   3500    3500#   3502    3502#   3504    3504#
          3506    3506#   3508    3508#   3510    3510#   3512    3512#   3514    3514#   3516    3516#   3518    3518#
          3520    3520#   3522#   3522    3527#   3527    3529    3529#   3531    3531#   3533    3533#   3535    3535#
          3537    3537#   3539    3539#   3541    3541#   3543    3543#   3548#   3548    3550    3550#   3552    3552#
          3554    3554#   3556    3556#   3558    3558#   3560    3560#   3562    3562#   3564    3564#   3734#   3734
          3736    3736#   3738    3738#   3740    3740#   3742    3742#   3744    3744#   3746    3746#   3748    3748#
          3750    3750#   3752    3752#   3754    3754#   3756    3756#   3761#   3761    3763    3763#   3765    3765#
          3767    3767#   3769    3769#   3771    3771#   3773    3773#   3775    3775#   3780#   3780    3782    3782#
          3784    3784#   3786    3786#   3788    3788#   3793#   3793    3798#   3798    3800    3800#   3802    3802#
          3804    3804#   3806    3806#   3808    3808#   3810    3810#   3812    3812#   3814    3814#   3816    3816#
          3818    3818#   3820    3820#   3822    3822#   3824    3824#   3826    3826#   3828    3828#   3830    3830#
          3832    3832#   3834    3834#   3836    3836#   3838    3838#   3840    3840#   3842    3842#   3844    3844#
          3846    3846#   3848    3848#   3850    3850#   3852#   3852    3857#   3857    3859    3859#   3861    3861#
          3863    3863#   3865    3865#   3867    3867#   3869    3869#   3871    3871#   3873    3873#   3878#   3878
          3880    3880#   3882    3882#   3884    3884#   3886    3886#   3888    3888#   3890    3890#   3892    3892#
          3894    3894#
%ML        142#    142     144#    144     146#    146     148#    148     150#    150     152#    152     154#    154
           156#    156     158#    158     163#    163     165#    165     167#    167     169#    169     171#    171
           173#    173     175#    175     177#    177     179#    179     184#    184     186#    185     188#    188
           190#    190     192#    192     194#    194     196#    196     198#    198     200#    200     205#    205
           207#    207     209#    209     211#    211     213#    213     215#    215     217#    217     219#    219
```

```
.MSE

.MSE
.MSK
.MSM
.MTI
.MXN

.RA

.RAN

.RB

.RC
.RC
.RC
.RC

.RC
.RC
.RC
.RC

.RC
```

```
 221#   221   360#   360   362#   362   364#   364   366#   366   368#   368   371#   371     SEQ 1981
 376#   376   378#   378   380#   380   382#   382   387#   387   389#   389   391#   391
 393#   393   398#   398   400#   400   406#   406   408#   408   410#   410   415#   415
 417#   417   419#   419   424#   424   426#   426   428#   428   430#   430   432#   432
 434#   434   436#   436   438#   438   440#   440   445#   445   447#   447   449#   449
 451#   451   453#   453   455#   455   457#   457   459#   459   461#   461   551#   551
 553#   553   555#   555   557#   557   559#   559   562#   562   564#   564   566#   566
 568#   568   570#   570   572#   572   662#   662   664#   664   666#   666   668#   668
 670#   670   673#   673   675#   675   677#   677   679#   679   681#   681   683#   683
 800#   800   802#   802   804#   804   806#   806   808#   808   813#   813   815#   815
 817#   817   819#   819   824#   824   826#   826   828#   828   830#   830   832#   832
 837#   837   839#   839   841#   841   843#   843   848#   848   853#   853   855#   855
 857#   857   859#   859   864#   864   866#   866   869#   869   874#   874   876#   876
 878#   878   880#   880   882#   882   884#   884   886#   886   888#   888   890#   890
 895#   895   897#   897   899#   899   901#   901   903#   903   905#   905   907#   907
 909#   909   911#   911  1051#  1051  1053#  1053  1055#  1055  1057#  1057  1059#  1059
1061#  1061  1063#  1063  1068#  1068  1070#  1070  1072#  1072  1077#  1077  1079#  1079
1081#  1081  1088#  1088  1090#  1090  1092#  1092  1094#  1094  1096#  1096  1098#  1098
1103#  1103  1105#  1105  1110#  1110  1115#  1115  1117#  1117  1240#  1240  1242#  1242
1247#  1247  1249#  1249  1251#  1251  1253#  1253  1255#  1255  1257#  1257  1259#  1259
1261#  1261  1266#  1266  1268#  1268  1270#  1270  1275#  1275  1280#  1280  1282#  1282
1459#  1459  1461#  1461  1463#  1463  1465#  1465  1467#  1467  1469#  1469  1471#  1471
1473#  1473  1479#  1479  1481#  1481  1483#  1483  1488#  1488  1490#  1490  1492#  1492
1499#  1499  1501#  1501  1503#  1503  1505#  1505  1507#  1507  1509#  1509  1514#  1514
1516#  1516  1521#  1521  1526#  1526  1528#  1528  1662#  1662  1664#  1664  1667#  1667
1669#  1669  1879#  1879  1881#  1881  1883#  1883  1885#  1885  1887#  1887  1889#  1889
1891#  1891  1896#  1896  1898#  1898  1975#  1975  1977#  1977  1979#  1979  1982#  1982
1984#  1984  2213#  2213  2215#  2215  2217#  2217  2219#  2219  2221#  2221  2223#  2223
2228#  2228  2230#  2230  2232#  2232  2234#  2234  2236#  2236  2238#  2238  2240#  2240
2242#  2242  2247#  2247  2254#  2254  2256#  2256  2258#  2258  2260#  2260  2262#  2262
2264#  2264  2266#  2266  2268#  2268  2273#  2273  2278#  2278  2283#  2283  2285#  2285
2290#  2290  2292#  2292  2294#  2294  2296#  2296  2298#  2298  2300#  2300  2302#  2302
2304#  2304  2306#  2306  2311#  2311  2313#  2313  2315#  2315  2317#  2317  2319#  2319
2321#  2321  2323#  2323  2325#  2325  2327#  2327  2542#  2542  2544#  2544  2546#  2546
2548#  2548  2550#  2550  2552#  2552  2557#  2557  2559#  2559  2561#  2561  2563#  2563
2565#  2565  2567#  2567  2569#  2569  2571#  2571  2576#  2576  2583#  2583  2585#  2585
2587#  2587  2589#  2589  2591#  2591  2593#  2593  2595#  2595  2597#  2597  2602#  2602
2607#  2607  2612#  2612  2614#  2614  2619#  2619  2621#  2621  2623#  2623  2625#  2625
2627#  2627  2629#  2629  2631#  2631  2633#  2633  2635#  2635  2640#  2640  2642#  2642
2644#  2644  2646#  2646  2648#  2648  2650#  2650  2652#  2652  2654#  2654  2656#  2656
2816#  2816  2818#  2818  2820#  2820  2822#  2822  2824#  2824  2826#  2826  2828#  2828
2830#  2830  2835#  2835  2837#  2837  2839#  2839  2841#  2841  2843#  2843  2845#  2845
2847#  2847  2849#  2849  2854#  2854  2856#  2856  2858#  2858  2860#  2860  2862#  2862
2867#  2867  2872#  2872  2874#  2874  2876#  2876  2878#  2878  2880#  2880  2882#  2882
2884#  2884  2886#  2886  2888#  2888  2890#  2890  2892#  2892  2894#  2894  2896#  2896
2898#  2898  2900#  2900  2902#  2902  2904#  2904  2906#  2906  2908#  2908  2910#  2910
2915#  2915  2917#  2917  2919#  2919  2921#  2921  2923#  2923  2925#  2925  2927#  2927
2929#  2929  2931#  2931  2936#  2936  2938#  2938  2940#  2940  2942#  2942  2944#  2944
2946#  2946  2948#  2948  2950#  2950  2952#  2952  3114#  3114  3116#  3116  3118#  3118
3120#  3120  3122#  3122  3124#  3124  3126#  3126  3128#  3128  3133#  3133  3135#  3135
3137#  3137  3139#  3139  3141#  3141  3143#  3143  3145#  3145  3147#  3147  3149#  3149
3154#  3154  3156#  3156  3158#  3158  3160#  3160  3162#  3162  3167#  3167  3172#  3172
3174#  3174  3176#  3176  3178#  3178  3180#  3180  3182#  3182  3184#  3184  3186#  3186
```

```
        3188#  3188    3190#  3190    3192#  3192    3194#  3194    3196#  3196    3198#  3198    3200#  3200    SEQ 1982
        3202#  3202    3204#  3204    3206#  3206    3208#  3208    3210#  3210    3215#  3215    3217#  3217
        3219#  3219    3221#  3221    3223#  3223    3225#  3225    3227#  3227    3229#  3229    3231#  3231
        3236#  3236    3238#  3238    3240#  3240    3242#  3242    3244#  3244    3246#  3246    3248#  3248
        3250#  3250    3252#  3252    3414#  3414    3416#  3416    3418#  3418    3420#  3420    3422#  3422
        3424#  3424    3426#  3426    3428#  3428    3430#  3430    3432#  3432    3437#  3437    3439#  3439
        3441#  3441    3443#  3443    3445#  3445    3447#  3447    3449#  3449    3451#  3451    3453#  3453
        3458#  3458    3460#  3460    3462#  3462    3464#  3464    3466#  3466    3471#  3471    3476#  3476
        3478#  3478    3480#  3480    3482#  3482    3484#  3484    3486#  3486    3488#  3488    3490#  3490
        3492#  3492    3494#  3494    3496#  3496    3498#  3498    3500#  3500    3502#  3502    3504#  3504
        3506#  3506    3508#  3508    3510#  3510    3512#  3512    3514#  3514    3516#  3516    3518#  3518
        3520#  3520    3522#  3522    3527#  3527    3529#  3529    3531#  3531    3533#  3533    3535#  3535
        3537#  3537    3539#  3539    3541#  3541    3543#  3543    3548#  3548    3550#  3550    3552#  3552
        3554#  3554    3556#  3556    3558#  3558    3560#  3560    3562#  3562    3564#  3564    3734#  3734
        3736#  3736    3738#  3738    3740#  3740    3742#  3742    3744#  3744    3746#  3746    3748#  3748
        3750#  3750    3752#  3752    3754#  3754    3756#  3756    3761#  3761    3763#  3763    3765#  3765
        3767#  3767    3769#  3769    3771#  3771    3773#  3773    3775#  3775    3780#  3780    3782#  3782
        3784#  3784    3786#  3786    3788#  3788    3793#  3793    3798#  3798    3800#  3800    3802#  3802
        3804#  3804    3806#  3806    3808#  3808    3810#  3810    3812#  3812    3814#  3814    3816#  3816
        3818#  3818    3820#  3820    3822#  3822    3824#  3824    3826#  3826    3828#  3828    3830#  3830
        3832#  3832    3834#  3834    3836#  3836    3838#  3838    3840#  3840    3842#  3842    3844#  3844
        3846#  3846    3848#  3848    3850#  3850    3852#  3852    3857#  3857    3859#  3859    3861#  3861
        3863#  3863    3865#  3865    3867#  3867    3869#  3869    3871#  3871    3873#  3873    3878#  3878
        3880#  3880    3882#  3882    3884#  3884    3886#  3886    3888#  3888    3890#  3890    3892#  3892
        3894#  3894
%MR      142#   142     144#   144     146#   146     148#   148     150#   150     152#   152     154#   154
         156#   156     158#   158     163#   163     165#   165     167#   167     169#   169     171#   171
         173#   173     175#   175     177#   177     179#   179     184#   184     186#   186     188#   188
         190#   190     192#   192     194#   194     196#   196     198#   198     200#   200     205#   205
         207#   207     209#   209     211#   211     213#   213     215#   215     217#   217     219#   219
         221#   221     360#   360     362#   362     364#   364     366#   366     368#   368     371#   371
         376#   376     378#   378     380#   380     382#   382     387#   387     389#   389     391#   391
         393#   393     398#   398     400#   400     406#   406     408#   408     410#   410     415#   415
         417#   417     419#   419     424#   424     426#   426     428#   428     430#   430     432#   432
         434#   434     436#   436     438#   438     440#   440     445#   445     447#   447     449#   449
         451#   451     453#   453     455#   455     457#   457     459#   459     461#   461     551#   551
         553#   553     555#   555     557#   557     559#   559     562#   562     564#   564     566#   566
         568#   568     570#   570     572#   572     662#   662     664#   664     666#   666     668#   668
         670#   670     673#   673     675#   675     677#   677     679#   679     681#   681     683#   683
         800#   800     802#   802     804#   804     806#   806     808#   808     813#   813     815#   815
         817#   817     819#   819     824#   824     826#   826     828#   828     830#   830     832#   832
         837#   837     839#   839     841#   841     843#   843     848#   848     853#   853     855#   855
         857#   857     859#   859     864#   864     866#   866     869#   869     874#   874     876#   876
         878#   878     880#   880     882#   882     884#   884     886#   886     888#   888     890#   890
         895#   895     897#   897     899#   899     901#   901     903#   903     905#   905     907#   907
         909#   909     911#   911    1051#  1051    1053#  1053    1055#  1055    1057#  1057    1059#  1059
        1061#  1061    1063#  1063    1068#  1068    1070#  1070    1072#  1072    1077#  1077    1079#  1079
        1081#  1081    1088#  1088    1090#  1090    1092#  1092    1094#  1094    1096#  1096    1098#  1098
        1103#  1103    1105#  1105    1110#  1110    1115#  1115    1117#  1117    1240#  1240    1242#  1242
        1247#  1247    1249#  1249    1251#  1251    1253#  1253    1255#  1255    1257#  1257    1259#  1259
        1261#  1261    1266#  1266    1268#  1268    1270#  1270    1275#  1275    1280#  1280    1282#  1282
        1459#  1459    1461#  1461    1463#  1463    1465#  1465    1467#  1467    1469#  1469    1471#  1471
        1473#  1473    1479#  1479    1481#  1481    1483#  1483    1488#  1488    1490#  1490    1492#  1492
```

```
1970      1499#  1499    1501#  1501    1503#  1503    1505#  1505    1507#  1507    1509#  1509    1514#  1514      SEQ 1983
          1516#  1516    1521#  1521    1526#  1526    1528#  1528    1662#  1662    1664#  1664    1667#  1667
          1669#  1669    1879#  1879    1881#  1881    1883#  1883    1885#  1885    1887#  1887    1889#  1889
          1891#  1891    1896#  1896    1898#  1898    1975#  1975    1977#  1977    1979#  1979    1982#  1982
          1984#  1984    2213#  2213    2215#  2215    2217#  2217    2219#  2219    2221#  2221    2223#  2223
          2228#  2228    2230#  2230    2232#  2232    2234#  2234    2236#  2236    2238#  2238    2240#  2240
          2242#  2242    2247#  2247    2254#  2254    2256#  2256    2258#  2258    2260#  2260    2262#  2262
          2264#  2264    2266#  2266    2268#  2268    2273#  2273    2278#  2278    2283#  2283    2285#  2285
          2290#  2290    2292#  2292    2294#  2294    2296#  2296    2298#  2298    2300#  2300    2302#  2302
          2304#  2304    2306#  2306    2311#  2311    2313#  2313    2315#  2315    2317#  2317    2319#  2319
          2321#  2321    2323#  2323    2325#  2325    2327#  2327    2542#  2542    2544#  2544    2546#  2546
          2548#  2548    2550#  2550    2552#  2552    2557#  2557    2559#  2559    2561#  2561    2563#  2563
          2565#  2565    2567#  2567    2569#  2569    2571#  2571    2576#  2576    2583#  2583    2585#  2585
          2587#  2587    2589#  2589    2591#  2591    2593#  2593    2595#  2595    2597#  2597    2602#  2602
          2607#  2607    2612#  2612    2614#  2614    2619#  2619    2621#  2621    2623#  2623    2625#  2625
          2627#  2627    2629#  2629    2631#  2631    2633#  2633    2635#  2635    2640#  2640    2642#  2642
          2644#  2644    2646#  2646    2648#  2648    2650#  2650    2652#  2652    2654#  2654    2656#  2656
          2816#  2816    2818#  2818    2820#  2820    2822#  2822    2824#  2824    2826#  2826    2828#  2828
          2830#  2830    2835#  2835    2837#  2837    2839#  2839    2841#  2841    2843#  2843    2845#  2845
          2847#  2847    2849#  2849    2854#  2854    2856#  2856    2858#  2858    2860#  2860    2862#  2862
          2867#  2867    2872#  2872    2874#  2874    2876#  2876    2878#  2878    2880#  2880    2882#  2882
          2884#  2884    2886#  2886    2888#  2888    2890#  2890    2892#  2892    2894#  2894    2896#  2896
          2898#  2898    2900#  2900    2902#  2902    2904#  2904    2906#  2906    2908#  2908    2910#  2910
          2915#  2915    2917#  2917    2919#  2919    2921#  2921    2923#  2923    2925#  2925    2927#  2927
          2929#  2929    2931#  2931    2936#  2936    2938#  2938    2940#  2940    2942#  2942    2944#  2944
          2946#  2946    2948#  2948    2950#  2950    2952#  2952    3114#  3114    3116#  3116    3118#  3118
          3120#  3120    3122#  3122    3124#  3124    3126#  3126    3128#  3128    3133#  3133    3135#  3135
          3137#  3137    3139#  3139    3141#  3141    3143#  3143    3145#  3145    3147#  3147    3149#  3149
          3154#  3154    3156#  3156    3158#  3158    3160#  3160    3162#  3162    3167#  3167    3172#  3172
          3174#  3174    3176#  3176    3178#  3178    3180#  3180    3182#  3182    3184#  3184    3186#  3186
          3188#  3188    3190#  3190    3192#  3192    3194#  3194    3196#  3196    3198#  3198    3200#  3200
          3202#  3202    3204#  3204    3206#  3206    3208#  3208    3210#  3210    3215#  3215    3217#  3217
          3219#  3219    3221#  3221    3223#  3223    3225#  3225    3227#  3227    3229#  3229    3231#  3231
          3236#  3236    3238#  3238    3240#  3240    3242#  3242    3244#  3244    3246#  3246    3248#  3248
          3250#  3250    3252#  3252    3414#  3414    3416#  3416    3418#  3418    3420#  3420    3422#  3422
          3424#  3424    3426#  3426    3428#  3428    3430#  3430    3432#  3432    3437#  3437    3439#  3439
          3441#  3441    3443#  3443    3445#  3445    3447#  3447    3449#  3449    3451#  3451    3453#  3453
          3458#  3458    3460#  3460    3462#  3462    3464#  3464    3466#  3466    3471#  3471    3476#  3476
          3478#  3478    3480#  3480    3482#  3482    3484#  3484    3486#  3486    3488#  3488    3490#  3490
          3492#  3492    3494#  3494    3496#  3496    3498#  3498    3500#  3500    3502#  3502    3504#  3504
          3506#  3506    3508#  3508    3510#  3510    3512#  3512    3514#  3514    3516#  3516    3518#  3518
          3520#  3520    3522#  3522    3527#  3527    3529#  3529    3531#  3531    3533#  3533    3535#  3535
          3537#  3537    3539#  3539    3541#  3541    3543#  3543    3548#  3548    3550#  3550    3552#  3552
          3554#  3554    3556#  3556    3558#  3558    3560#  3560    3562#  3562    3564#  3564    3734#  3734
          3736#  3736    3738#  3738    3740#  3740    3742#  3742    3744#  3744    3746#  3746    3748#  3748
          3750#  3750    3752#  3752    3754#  3754    3756#  3756    3761#  3761    3763#  3763    3765#  3765
          3767#  3767    3769#  3769    3771#  3771    3773#  3773    3775#  3775    3780#  3780    3782#  3782
          3784#  3784    3786#  3786    3788#  3788    3793#  3793    3798#  3798    3800#  3800    3802#  3802
          3804#  3804    3806#  3806    3808#  3808    3810#  3810    3812#  3812    3814#  3814    3816#  3816
          3818#  3818    3820#  3820    3822#  3822    3824#  3824    3826#  3826    3828#  3828    3830#  3830
          3832#  3832    3834#  3834    3836#  3836    3838#  3838    3840#  3840    3842#  3842    3844#  3844
          3846#  3846    3848#  3848    3850#  3850    3852#  3852    3857#  3857    3859#  3859    3861#  3861
          3863#  3863    3865#  3865    3867#  3867    3869#  3869    3871#  3871    3873#  3873    3878#  3878
          3880#   3880  3882#   3882  3884#   3884  3886#   3886  3888#   3888  3890#   3890  3892#   3892   SEQ 1984
          3894#   3894
.LA         156    177    190    192    198    200    211    213    219    221    371    376    380    391
            415    417    419    430    432    438    451    453    459    559    572    670    683    802
            806    813    817    826    830    837    841    880    882    888    901    903    909   1070
           1079   1103   1242   1253   1268   1481   1490   1514   2230   2240   2273   2278   2296   2298
           2304   2317   2319   2325   2559   2569   2602   2607   2625   2627   2633   2646   2648   2654
           2837   2841   2878   2882   2886   2890   2894   2898   2902   2906   2921   2923   2929   2942
           2944   2950   3135   3141   3178   3182   3186   3190   3194   3198   3202   3206   3221   3223
           3229   3242   3244   3250   3439   3445   3482   3486   3490   3494   3498   3502   3506   3510
           3514   3518   3533   3535   3541   3554   3556   3562   3763   3767   3804   3808   3812   3816
           3820   3824   3828   3832   3836   3840   3844   3848   3863   3865   3871   3884   3886   3892
.LADDR      142    163    184    205    360    406    415    424    445    551    562    662    673    800
            853    864    874    895   1051   1088   1103   1110   1115   1240   1275   1280   1459   1499
           1514   1521   1526   1662   1667   1879   1896   1975   2213   2215   2254   2273   2278   2283
           2290   2311   2542   2544   2583   2602   2607   2612   2619   2640   2816   2835   2854   2867
           2872   2910   2915   2936   3114   3133   3154   3167   3172   3210   3215   3236   3414   3437
           3458   3471   3476   3522   3527   3548   3734   3761   3780   3793   3798   3852   3857   3878
.LAND       184    205    362    406    408    424    445    551    562    662    673    800    874    895
           1051   1240   1459   1662   1667   1984   2219   2290   2311   2548   2619   2640   2915   2936
           3215   3236   3527   3548   3857   3878
.LB         148    154    156    169    175    177    184    186    190    205    207    211    360    362
            364    366    368    376    378    380    387    389    391    406    408    410    424    426
            430    445    447    451    557    570    668    681    800    804    815    824    828    839
            874    876    880    895    897    901   1051   1096   1098   1240   1242   1255   1257   1259
           1459   1507   1509   1887   2219   2240   2266   2290   2292   2296   2311   2313   2317   2548
           2569   2595   2619   2621   2625   2640   2642   2646   2816   2818   2820   2822   2824   2826
           2828   2830   2862   2915   2917   2921   2936   2938   2942   3114   3116   3118   3120   3122
           3124   3126   3128   3162   3215   3217   3221   3236   3238   3242   3414   3416   3418   3420
           3422   3424   3426   3428   3430   3432   3466   3527   3529   3533   3548   3550   3554   3734
           3736   3738   3740   3742   3744   3746   3748   3750   3752   3754   3756   3788   3857   3859
           3863   3878   3880   3884
.LCCCA      855   1090   1249   1501   1881   2260   2589   2856   3156   3460   3782
.LCCCP     1059   1467
.LCCFZ      156    177    380    391    806    817    830    841   1259
.LCCGC      194    215    434    455    884    905   2300   2321   2629   2650   2925   2946   3225   3246
           3537   3558   3867   3888
.LCCLW     1061   1471   2240   2569
.LCCMB     2230   2559   2841   3141   3445   3767
.LCCMP     2232   2236   2561   2565   2843   2847   3143   3147   3447   3451   3769   3773
.LCENA      156    177    194    215    380    391    434    455    806    817    830    841    855    884
            905   1059   1061   1090   1249   1259   1467   1471   1501   1881   2230   2232   2236   2240
           2260   2300   2321   2559   2561   2565   2569   2589   2629   2650   2841   2843   2847   2856
           2925   2946   3141   3143   3147   3156   3225   3246   3445   3447   3451   3460   3537   3558
           3767   3769   3773   3782   3867   3888
.LCJP       156    177    194    215    380    391    434    455    806    817    830    841    855    884
            905   1059   1061   1090   1249   1259   1467   1471   1501   1881   2230   2232   2236   2240
           2260   2300   2321   2559   2561   2565   2569   2589   2629   2650   2841   2843   2847   2856
           2925   2946   3141   3143   3147   3156   3225   3246   3445   3447   3451   3460   3537   3558
           3767   3769   3773   3782   3867   3888
.LCJS       800    824   1053   1057   1461   1465   2221   2228   2550   2557   2876   2880   2884   2888
           2892   2896   2900   2904   3176   3180   3184   3188   3192   3196   3200   3204   3480   3484
           3488   3492   3496   3500   3504   3508   3512   3516   3802   3806   3810   3814   3818   3822
          3826  3830  3834  3838  3842  3846
.LCONT     148   150   152   154   169   171   173   175   184   186   192   198   205   207     SEQ 1985
           213   219   364   371   376   378   387   389   408   424   426   432   438   445
           447   453   459   553   555   557   564   566   568   570   664   666   668   675
           677   679   681   802   804   813   815   826   828   837   839   853   874   876
           882   888   895   897   903   909  1068  1070  1077  1079  1088  1094  1096  1242
          1247  1253  1255  1257  1266  1479  1481  1488  1490  1499  1505  1507  1885  1887
          1889  1977  2219  2254  2264  2266  2290  2292  2298  2304  2311  2313  2319  2325
          2548  2583  2593  2595  2619  2621  2627  2633  2640  2642  2648  2654  2835  2837
          2839  2860  2874  2878  2882  2886  2890  2894  2898  2902  2906  2915  2917  2923
          2929  2936  2938  2944  2950  3133  3135  3137  3139  3160  3174  3178  3182  3186
          3190  3194  3198  3202  3206  3215  3217  3223  3229  3236  3238  3244  3250  3437
          3439  3441  3443  3464  3478  3482  3486  3490  3494  3498  3502  3506  3510  3514
          3518  3527  3529  3535  3541  3548  3550  3556  3562  3761  3763  3765  3786  3800
          3804  3808  3812  3816  3820  3824  3828  3832  3836  3840  3844  3848  3857  3859
          3865  3871  3878  3880  3886  3892
.LCRTN     859  1098  1509  2268  2597  2862  3162  3466  3788
.LCRY      156   177   186   207   364   368   380   391   410   426   447   876   897  1098
          1242  1255  1509  1664  1669  2240  2292  2313  2569  2621  2642  2917  2938  3217
          3238  3529  3550  38[illegible]  3880
.LD        142   144   146  [illegible]8   150   152   154   156   158   163   165   167   169   171
           173   175   177   179   184   186   190   192   194   196   198   200   205   207
           211   213   215   217   219   221   360   362   364   366   368   371   376   378
           382   387   389   393   398   400   406   408   410   415   417   419   424   426
           430   432   434   436   438   440   445   447   451   453   455   457   459   461
           551   553   555   557   559   562   564   566   568   570   572   662   664   666
           668   670   673   675   677   679   681   683   800   802   804   806   808   813
           815   817   819   824   826   828   830   832   837   839   841   843   848   853
           855   857   859   874   876   880   882   884   886   888   890   895   897   901
           903   905   907   909   911  1051  1053  1055  1057  1059  1061  1070  1079  1088
          1090  1092  1094  1096  1098  1103  1240  1242  1247  1249  1251  1253  1255  1257
          1259  1268  1459  1461  1463  1465  1467  1469  1471  1481  1490  1499  1501  1503
          1505  1507  1509  1514  1662  1664  1667  1669  1881  1883  1885  1887  1975  1977
          1979  1984  2219  2221  2228  2230  2232  2234  2236  2238  2240  2254  2256  2258
          2260  2262  2264  2266  2268  2273  2278  2290  2292  2296  2298  2300  2302  2304
          2306  2311  2313  2317  2319  2321  2323  2325  2327  2548  2550  2557  2559  2561
          2563  2565  2567  2569  2583  2585  2587  2589  2591  2593  2595  2597  2602  2607
          2619  2621  2625  2627  2629  2631  2633  2635  2640  2642  2646  2648  2650  2652
          2654  2656  2816  2818  2820  2822  2824  2826  2828  2830  2835  2837  2839  2841
          2843  2845  2847  2849  2854  2856  2858  2860  2862  2876  2878  2880  2882  2884
          2886  2888  2890  2892  2894  2896  2898  2900  2902  2904  2906  2915  2917  2921
          2923  2925  2927  2929  2931  2936  2938  2942  2944  2946  2948  2950  2952  3114
          3116  3118  3120  3122  3124  3126  3128  3133  3135  3137  3139  3141  3143  3145
          3147  3149  3154  3156  3158  3160  3162  3176  3178  3180  3182  3184  3186  3188
          3190  3192  3194  3196  3198  3200  3202  3204  3206  3215  3217  3221  3223  3225
          3227  3229  3231  3236  3238  3242  3244  3246  3248  3250  3252  3414  3416  3418
          3420  3422  3424  3426  3428  3430  3432  3437  3439  3441  3443  3445  3447  3449
          3451  3453  3458  3460  3462  3464  3466  3480  3482  3484  3486  3488  3490  3492
          3494  3496  3498  3500  3502  3504  3506  3508  3510  3512  3514  3516  3518  3527
          3529  3533  3535  3537  3539  3541  3543  3548  3550  3554  3556  3558  3560  3562
          3564  3734  3736  3738  3740  3742  3744  3746  3748  3750  3752  3754  3756  3761
          3763  3765  3767  3769  3771  3773  3775  3780  3782  3784  3786  3788  3802  3804
          3806  3808  3810  3812  3814  3816  3818  3820  3822  3824  3826  3828  3830  3832
```

```
Q 1973          3834  3836  3838  3840  3842  3844  3846  3848  3857  3859  3863  3865  3867  3869  SEQ 1986
                3871  3873  3878  3880  3884  3886  3888  3890  3892  3894
       .LJ       142   144   146   156   158   163   165   167   177   179   188   190   194   196
                 200   209   211   215   217   221   360   362   366   368   380   382   391   393
                 398   400   406   410   415   417   419   428   430   434   436   440   449   451
                 455   457   461   551   559   562   572   662   670   673   683   800   806   808
                 817   819   824   830   832   841   843   848   855   857   864   866   878   880
                 884   886   890   899   901   905   907   911  1051  1053  1055  1057  1059  1061
                1063  1072  1081  1090  1092  1103  1105  1115  1117  1240  1249  1251  1259  1261
                1268  1270  1280  1282  1459  1461  1463  1465  1467  1469  1471  1473  1483  1492
                1501  1503  1514  1516  1526  1528  1662  1664  1667  1669  1879  1881  1883  1891
                1896  1898  1975  1979  1982  1984  2215  2217  2221  2223  2228  2230  2232  2234
                2236  2238  2240  2242  2247  2256  2258  2260  2262  2273  2278  2283  2285  2294
                2296  2300  2302  2306  2315  2317  2321  2323  2327  2544  2546  2550  2552  2557
                2559  2561  2563  2565  2567  2569  2571  2576  2585  2587  2589  2591  2602  2607
                2612  2614  2623  2625  2629  2631  2635  2644  2646  2650  2652  2656  2816  2818
                2820  2822  2824  2826  2828  2830  2841  2843  2845  2847  2849  2854  2856  2858
                2872  2876  2880  2884  2888  2892  2896  2900  2904  2919  2921  2925  2927  2931
                2940  2942  2946  2948  2952  3114  3116  3118  3120  3122  3124  3126  3128  3141
                3143  3145  3147  3149  3154  3156  3158  3172  3176  3180  3184  3188  3192  3196
                3200  3204  3219  3221  3225  3227  3231  3240  3242  3246  3248  3252  3414  3416
                3418  3420  3422  3424  3426  3428  3430  3432  3445  3447  3449  3451  3453  3458
                3460  3462  3476  3480  3484  3488  3492  3496  3500  3504  3508  3512  3516  3531
                3533  3537  3539  3543  3552  3554  3558  3560  3564  3734  3736  3738  3740  3742
                3744  3746  3748  3750  3752  3754  3756  3767  3769  3771  3773  3775  3780  3782
                3784  3798  3802  3806  3810  3814  3818  3822  3826  3830  3834  3838  3842  3846
                3861  3863  3867  3869  3873  3882  3884  3888  3890  3894
       .LJMAP    142   158   163   179   196   200   217   221   360   382   393   400   406   410
                 415   417   419   436   440   457   461   551   559   562   572   662   670   673
                 683   808   819   832   843   848   857   864   866   886   890   907   911  1051
                1055  1063  1072  1081  1092  1103  1105  1115  1117  1240  1251  1261  1268  1270
                1280  1282  1459  1463  1469  1473  1483  1492  1503  1514  1516  1526  1528  1662
                1664  1667  1669  1879  1883  1891  1896  1898  1975  1979  1982  1984  2223  2234
                2238  2242  2247  2262  2273  2278  2283  2285  2302  2306  2323  2327  2552  2563
                2567  2571  2576  2591  2602  2607  2612  2614  2631  2635  2652  2656  2816  2818
                2820  2822  2824  2826  2828  2830  2845  2849  2854  2858  2872  2927  2931  2948
                2952  3114  3116  3118  3120  3122  3124  3126  3128  3145  3149  3154  3158  3172
                3227  3231  3248  3252  3414  3416  3418  3420  3422  3424  3426  3428  3430  3432
                3449  3453  3458  3462  3476  3539  3543  3560  3564  3734  3736  3738  3740  3742
                3744  3746  3748  3750  3752  3754  3756  3771  3775  3780  3784  3798  3869  3873
                3890  3894
       .LJZ      869  1110  1275  1521  2213  2542  2867  2908  2910  3167  3208  3210  3471  3520
                3522  3793  3850  3852
       .LLDCT    144   165   188   209   362   368   428   449   878   899  2215  2256  2294  2315
                2544  2585  2623  2644  2919  2940  3219  3240  3531  3552  3861  3882
       .LLDLM    371  2878  2882  2886  2890  2894  2898  2902  2906  3178  3182  3186  3190  3194
                3198  3202  3206  3482  3486  3490  3494  3498  3502  3506  3510  3514  3518  3604
                3808  3812  3816  3820  3824  3828  3832  3836  3840  3844  3848
       .LMGC     144   148   150   152   154   165   169   171   173   175   192   194   198   200
                 213   215   219   221   360   371   376   378   387   389   415   417   419   432
                 434   438   453   455   459   551   553   555   557   559   564   565   568   570
                 572   662   664   666   668   670   675   677   679   681   683   802   804   813
                 815   826   828   837   839   853   864   866   882   884   888   903   905   909
```

```
EQ 1974   1068  1070  1077  1079  1088  1094  1096  1103  1110  1115  1117  1247  1253  1255   SEQ 1987
          1266  1268  1275  1280  1282  1479  1481  1488  1490  1499  1505  1507  1514  1521
          1526  1528  1664  1669  1879  1885  1887  1889  1896  1898  1975  1977  1979  1982
          1984  2213  2247  2254  2264  2266  2273  2278  2283  2285  2298  2300  2304  2319
          2321  2325  2542  2576  2583  2593  2595  2602  2607  2612  2614  2627  2629  2633
          2648  2650  2654  2816  2818  2820  2822  2824  2826  2828  2830  2835  2837  2839
          2854  2860  2862  2867  2872  2874  2878  2882  2886  2890  2894  2898  2902  2906
          2923  2925  2929  2944  2946  2950  3114  3116  3118  3120  3122  3124  3126  3128
          3133  3135  3137  3139  3154  3160  3162  3167  3172  3174  3178  3182  3186  3190
          3194  3198  3202  3206  3223  3225  3229  3244  3246  3250  3414  3416  3418  3420
          3422  3424  3426  3428  3430  3432  3437  3439  3441  3443  3458  3464  3466  3471
          3476  3478  3482  3486  3490  3494  3498  3502  3506  3510  3514  3518  3535  3537
          3541  3556  3558  3562  3734  3736  3738  3740  3742  3744  3746  3748  3750  3752
          3754  3756  3761  3763  3765  3780  3786  3788  3793  3798  3800  3804  3808  3812
          3816  3820  3824  3828  3832  3836  3840  3844  3848  3865  3867  3871  3886  3888
          3892
.LOENA     148   150   169   171   192   200   213   221   371   376   415   417   419   432
           453   551   553   559   564   566   572   662   664   670   675   677   683   802
           813   826   837   882   903  1070  1079  1103  1253  1268  1481  1490  1514  1664
          1669  1979  1984  2273  2278  2298  2319  2602  2607  2627  2648  2837  2878  2882
          2886  2890  2894  2898  2902  2906  2923  2944  3135  3178  3182  3186  3190  3194
          3198  3202  3206  3223  3244  3439  3482  3486  3490  3494  3498  3502  3506  3510
          3514  3518  3535  3556  3763  3804  3808  3812  3816  3820  3824  3828  3832  3836
          3840  3844  3848  3865  3886
.LOR       144   146   150   154   165   167   171   175   192   198   200   213   219   221
           360   366   376   378   387   389   415   417   419   432   438   453   459   557
           570   668   681   802   804   806   813   815   826   828   839   841   882   888
           903   909  1079  1096  1103  1253  1257  1259  1268  1490  1507  1514  1887  1977
          1979  2230  2266  2273  2278  2298  2304  2319  2325  2559  2595  2602  2607  2627
          2633  2648  2654  2816  2818  2820  2822  2824  2826  2828  2830  2862  2923  2929
          2944  2950  3114  3116  3118  3120  3122  3124  3126  3128  3162  3223  3229  3244
          3250  3414  3416  3418  3420  3422  3424  3426  3428  3430  3432  3466  3535  3541
          3556  3562  3734  3736  3738  3740  3742  3744  3746  3748  3750  3752  3754  3756
          3788  3865  3871  3886  3892
.LPLUS     186   190   207   211   364   368   410   426   430   447   451   876   880   897
           901  1098  1242  1255  1509  1664  1669  2240  2292  2296  2313  2317  2569  2621
          2625  2642  2646  2917  2921  2938  2942  3217  3221  3238  3242  3529  3533  3550
          3554  3859  3863  3880  3884
.LRDLM     387  2816  2818  2820  2822  2824  2826  2828  2830  3114  3116  3118  3120  3122
          3124  3126  3128  3414  3416  3418  3420  3422  3424  3426  3428  3430  3432  3734
          3736  3738  3740  3742  3744  3746  3748  3750  3752  3754  3756
.LRMIN     156   177
.LRPCT     146   167   190   211   366   398   430   451   880   901  2217  2258  2296  2317
          2546  2587  2625  2646  2921  2942  3221  3242  3533  3554  3863  3884
.LSOA      146   148   150   167   169   171   192   198   200   213   219   221   362   371
           376   406   408   415   417   419   432   438   453   459   551   553   559   562
           564   566   572   662   664   670   673   675   677   683   800   802   806   813
           817   826   830   837   841   882   888   903   909  1070  1079  1103  1242  1253
          1268  1481  1490  1514  1662  1667  1979  1984  2230  2240  2273  2278  2298  2304
          2319  2325  2559  2569  2602  2607  2627  2633  2648  2654  2837  2841  2878  2882
          2886  2890  2894  2898  2902  2906  2923  2929  2944  2950  3135  3141  3178  3182
          3186  3190  3194  3198  3202  3206  3223  3229  3244  3250  3439  3445  3482  3486
          3490  3494  3498  3502  3506  3510  3514  3518  3535  3541  3556  3562  3763  3767
```

```
EQ 1975                                                                                   SEQ 1988
          3804 3808 3812 3816 3820 3824 3828 3832 3836 3840 3844 3848 3865 3871
          3886 3892
.LSOB      186  207  364  366  368  410  426  447  824  876  897 1051 1098 1240
          1255 1257 1259 1459 1509 1664 1669 2219 2292 2313 2548 2621 2642 2917
          2938 3217 3238 3529 3550 3859 3880
.LSAB      156  177  190  211  380  391  430  451  880  901 2296 2317 2625 2646
          2921 2942 3221 3242 3533 3554 3863 3884
.LSDO      144  152  154  165  173  175  184  205  360  378  387  389  424  445
           555  557  568  570  666  668  679  681  804  815  828  839  874  895
          1096 1507 1887 1977 2266 2290 2311 2595 2619 2640 2816 2818 2820 2822
          2824 2826 2828 2830 2862 2915 2936 3114 3116 3118 3120 3122 3124 3126
          3128 3137 3162 3215 3236 3414 3416 3418 3420 3422 3424 3426 3428 3430
          3432 3441 3466 3527 3548 3734 3736 3738 3740 3742 3744 3746 3748 3750
          3752 3754 3756 3788 3857 3878
.LSELC     813  837  853  864 1068 1077 1088 1094 1115 1247 1255 1266 1280 1479
          1488 1499 1505 1526 1879 1885 1889 1896 1975 1982 2247 2254 2264 2283
          2576 2583 2593 2612 2854 2860 2867 2872 3154 3160 3167 3172 3458 3464
          3471 3476 3765 3780 3786 3793 3798
.LSELE     192  194  198  200  213  215  219  221  415  417  419  432  434  438
           453  455  459  559  572  670  683  882  884  888  903  905  909 1070
          1079 1103 1268 1481 1490 1514 1664 1669 1979 1984 2273 2278 2298 2300
          2304 2319 2321 2325 2602 2607 2627 2629 2633 2648 2650 2654 2923 2925
          2929 2944 2946 2950 3223 3225 3229 3244 3246 3250 3535 3537 3541 3556
          3558 3562 3865 3867 3871 3886 3888 3892
.LSELF     148  169  376  387  551  564  662  675  802  826 1253 2837 2839 3139
          3443 3763
.LSELM     150  154  171  175  378  389  553  557  566  570  664  668  677  681
           804  815  828  839  866 1096 1110 1117 1275 1282 1507 1521 1528 1887
          1898 1977 2213 2266 2285 2542 2595 2614 2835 2862 2874 3133 3135 3162
          3174 3437 3439 3466 3478 3761 3788 3800
.LSELP     152  173  555  568  666  679 3137 3441
.LSKCN     144  165  360
.LSMIN     380  391
.LTIM     1667 1669
.LXNOR     148  169  371  553  564  566  664  675  677  817  824  830  837 1070
          1481
.MA        156  177  190  192  198  200  211  213  219  221  371  376  380  391
           415  417  419  430  432  438  451  453  459  559  572  670  683  802
           806  813  817  826  830  837  841  880  882  888  901  903  909 1070
          1079 1103 1242 1253 1268 1481 1490 1514 2230 2240 2273 2278 2296 2298
          2304 2317 2319 2325 2559 2569 2602 2607 2625 2627 2633 2646 2648 2654
          2837 2841 2878 2882 2886 2890 2894 2898 2902 2906 2921 2923 2929 2942
          2944 2950 3135 3141 3178 3182 3186 3190 3194 3198 3202 3206 3221 3223
          3229 3242 3244 3250 3439 3445 3482 3486 3490 3494 3498 3502 3506 3510
          3514 3518 3533 3535 3541 3554 3556 3562 3763 3767 3804 3808 3812 3816
          3820 3824 3828 3832 3836 3840 3844 3848 3863 3865 3871 3884 3886 3892
.MAND      184  205  362  406  408  424  445  551  562  662  673  800  874  895
          1051 1240 1459 1662 1667 1984 2219 2290 2311 2548 2619 2640 2915 2936
          3215 3236 3527 3548 3857 3878
.MB        148  154  156  169  175  177  184  186  190  205  207  211  360  362
           364  366  368  376  378  380  387  389  391  406  408  410  424  426
           430  445  447  451  557  570  668  681  800  804  815  824  828  839
           874  876  880  895  897  901 1051 1096 1098 1240 1242 1255 1257 1259
```

```
SEQ 1976                                                                                     SEQ 1989

         1459  1507  1509  1887  2219  2240  2266  2290  2292  2296  2311  2313  2317  2548
         2569  2595  2619  2621  2625  2640  2642  2646  2816  2818  2820  2822  2824  2826
         2828  2830  2862  2915  2917  2921  2936  2938  2942  3114  3116  3118  3120  3122
         3124  3126  3128  3162  3215  3217  3221  3236  3238  3242  3414  3416  3418  3420
         3422  3424  3426  3428  3430  3432  3466  3527  3529  3533  3548  3550  3554  3734
         3736  3738  3740  3742  3744  3746  3748  3750  3752  3754  3756  3788  3857  3859
         3863  3878  3880  3884
.MCCCA    855  1090  1249  1501  1881  2260  2589  2856  3156  3460  3782
.MCCCP   1059  1467
.MCCFZ    156   177   380   391   806   817   830   841  1259
.MCCGC    194   215   434   455   884   905  2300  2321  2629  2650  2925  2946  3225  3246
         3537  3558  3867  3888
.MCCLW   1061  1471  2240  2569
.MCCMB   2230  2559  2841  3141  3445  3767
.MCCMP   2232  2236  2561  2565  2843  2847  3143  3147  3447  3451  3769  3773
.MCENA    156   177   194   215   380   391   434   455   806   817   830   841   855   884
          905  1059  1061  1090  1249  1259  1467  1471  1501  1881  2230  2232  2236  2240
         2260  2300  2321  2559  2561  2565  2569  2589  2629  2650  2841  2843  2847  2856
         2925  2946  3141  3143  3147  3156  3225  3246  3445  3447  3451  3460  3537  3558
         3767  3769  3773  3782  3867  3888
.MCJP     156   177   194   215   380   391   434   455   806   817   830   841   855   884
          905  1059  1061  1090  1249  1259  1467  1471  1501  1881  2230  2232  2236  2240
         2260  2300  2321  2559  2561  2565  2569  2589  2629  2650  2841  2843  2847  2856
         2925  2946  3141  3143  3147  3156  3225  3246  3445  3447  3451  3460  3537  3558
         3767  3769  3773  3782  3867  3888
.MCJS     800   824  1053  1057  1461  1465  2221  2228  2550  2557  2876  2880  2884  2888
         2892  2896  2900  2904  3176  3180  3184  3188  3192  3196  3200  3204  3480  3484
         3488  3492  3496  3500  3504  3508  3512  3516  3802  3806  3810  3814  3818  3822
         3826  3830  3834  3838  3842  3846
.MCONT    148   150   152   154   169   171   173   175   184   186   192   198   205   207
          213   219   364   371   376   378   387   389   408   424   426   432   438   445
          447   453   459   553   555   557   564   566   568   570   664   666   668   675
          677   679   681   802   804   813   815   826   828   837   839   853   874   876
          882   888   895   897   903   909  1068  1070  1077  1079  1088  1094  1096  1242
         1247  1253  1255  1257  1266  1479  1481  1488  1490  1499  1505  1507  1885  1887
         1889  1977  2219  2254  2264  2266  2290  2292  2298  2304  2311  2313  2319  2325
         2548  2583  2593  2595  2619  2621  2627  2633  2640  2642  2648  2654  2835  2837
         2839  2860  2874  2878  2882  2886  2890  2894  2898  2902  2906  2915  2917  2923
         2929  2936  2938  2944  2950  3133  3135  3137  3139  3160  3174  3178  3182  3186
         3190  3194  3198  3202  3206  3215  3217  3223  3229  3236  3238  3244  3250  3437
         3439  3441  3443  3464  3478  3482  3486  3490  3494  3498  3502  3506  3510  3514
         3518  3527  3529  3535  3541  3548  3550  3556  3562  3761  3763  3765  3786  3800
         3804  3808  3812  3816  3820  3824  3828  3832  3836  3840  3844  3848  3857  3859
         3865  3871  3878  3880  3886  3892
.MCRTN    859  1098  1509  2268  2597  2862  3162  3466  3788
.MCRY     156   177   186   207   364   368   380   391   410   426   447   876   897  1098
         1242  1255  1509  1664  1669  2240  2292  2313  2569  2621  2642  2917  2938  3217
         3238  3529  3550  3859  3880
.MD       142   144   146   148   150   152   154   156   158   163   165   167   169   171
          173   175   177   179   184   186   190   192   194   196   198   200   205   207
          211   213   215   217   219   221   360   362   364   366   368   371   376   378
          382   387   389   393   398   400   406   408   410   415   417   419   424   426
          430   432   434   436   438   440   445   447   451   453   455   457   459   461
```

977

```
          551   553   555   557   559   562   564   566   568   570   572   662   664   666     SEQ 1990
          668   670   673   675   677   679   681   683   800   802   804   806   808   813
          815   817   819   824   826   828   830   832   837   839   841   843   848   853
          855   857   859   874   876   880   882   884   886   888   890   895   897   901
          903   905   907   909   911  1051  1053  1055  1057  1059  1061  1070  1079  1088
         1090  1092  1094  1096  1098  1103  1240  1242  1247  1249  1251  1253  1255  1257
         1259  1268  1459  1461  1463  1465  1467  1469  1471  1481  1490  1499  1501  1503
         1505  1507  1509  1514  1662  1664  1667  1669  1881  1883  1885  1887  1975  1977
         1979  1984  2219  2221  2228  2230  2232  2234  2236  2238  2240  2254  2256  2258
         2260  2262  2264  2266  2268  2273  2278  2290  2292  2296  2298  2300  2302  2304
         2306  2311  2313  2317  2319  2321  2323  2325  2327  2548  2550  2557  2559  2561
         2563  2565  2567  2569  2583  2585  2587  2589  2591  2593  2595  2597  2602  2607
         2619  2621  2625  2627  2629  2631  2633  2635  2640  2642  2646  2648  2650  2652
         2654  2656  2816  2818  2820  2822  2824  2826  2828  2830  2835  2837  2839  2841
         2843  2845  2847  2849  2854  2856  2858  2860  2862  2876  2878  2880  2882  2884
         2886  2888  2890  2892  2894  2896  2898  2900  2902  2904  2906  2915  2917  2921
         2923  2925  2927  2929  2931  2936  2938  2942  2944  2946  2948  2950  2952  3114
         3116  3118  3120  3122  3124  3126  3128  3133  3135  3137  3139  3141  3143  3145
         3147  3149  3154  3156  3158  3160  3162  3176  3178  3180  3182  3184  3186  3188
         3190  3192  3194  3196  3198  3200  3202  3204  3206  3215  3217  3221  3223  3225
         3227  3229  3231  3236  3238  3242  3244  3246  3248  3250  3252  3414  3416  3418
         3420  3422  3424  3426  3428  3430  3432  3437  3439  3441  3443  3445  3447  3449
         3451  3453  3458  3460  3462  3464  3466  3480  3482  3484  3486  3488  3490  3492
         3494  3496  3498  3500  3502  3504  3506  3508  3510  3512  3514  3516  3518  3527
         3529  3533  3535  3537  3539  3541  3543  3548  3550  3554  3556  3558  3560  3562
         3564  3734  3736  3738  3740  3742  3744  3746  3748  3750  3752  3754  3756  3761
         3763  3765  3767  3769  3771  3773  3775  3780  3782  3784  3786  3788  3802  3804
         3806  3808  3810  3812  3814  3816  3818  3820  3822  3824  3826  3828  3830  3832
         3834  3836  3838  3840  3842  3844  3846  3848  3857  3859  3863  3865  3867  3869
         3871  3873  3878  3880  3884  3886  3888  3890  3892  3894
.MJ       142   144   146   156   158   163   165   167   177   179   188   190   194   196
          200   209   211   215   217   221   360   362   366   368   380   382   391   393
          398   400   406   410   415   417   419   428   430   434   436   440   449   451
          455   457   461   551   559   562   572   662   670   673   683   800   806   808
          817   819   824   830   832   841   843   848   855   857   864   866   878   880
          884   886   890   899   901   905   907   911  1051  1053  1055  1057  1059  1061
         1063  1072  1081  1090  1092  1103  1105  1115  1117  1240  1249  1251  1259  1261
         1268  1270  1280  1282  1459  1461  1463  1465  1467  1469  1471  1473  1483  1492
         1501  1503  1514  1516  1526  1528  1662  1664  1667  1669  1879  1881  1883  1891
         1896  1898  1975  1979  1982  1984  2215  2217  2221  2223  2228  2230  2232  2234
         2236  2238  2240  2242  2247  2256  2258  2260  2262  2273  2278  2283  2285  2294
         2296  2300  2302  2306  2315  2317  2321  2323  2327  2544  2546  2550  2552  2557
         2559  2561  2563  2565  2567  2569  2571  2576  2585  2587  2589  2591  2602  2607
         2612  2614  2623  2625  2629  2631  2635  2644  2646  2650  2652  2656  2816  2818
         2820  2822  2824  2826  2828  2830  2841  2843  2845  2847  2849  2854  2856  2858
         2872  2876  2880  2884  2888  2892  2896  2900  2904  2919  2921  2925  2927  2931
         2940  2942  2946  2948  2952  3114  3116  3118  3120  3122  3124  3126  3128  3141
         3143  3145  3147  3149  3154  3156  3158  3172  3176  3180  3184  3188  3192  3196
         3200  3204  3219  3221  3225  3227  3231  3240  3242  3246  3248  3252  3414  3416
         3418  3420  3422  3424  3426  3428  3430  3432  3445  3447  3449  3451  3453  3458
         3460  3462  3476  3480  3484  3488  3492  3496  3500  3504  3508  3512  3516  3531
         3533  3537  3539  3543  3552  3554  3558  3560  3564  3734  3736  3738  3740  3742
         3744  3746  3748  3750  3752  3754  3756  3767  3769  3771  3773  3775  3780  3782
```

```
PDEC
PNT3
PNTDE
PNTHW
PNTMS

PNTOC
RTN

SCOPE

TMSG

TMSGC

TTABL
```

```
         3798  3802  3806  3810  3814  3818  3822  3826  3830  3834  3838  3842  3846
```

```
           3784  3798  3802  3806  3810  3814  3818  3822  3826  3830  3834  3838  3842  3846     SEQ 1991
           3861  3863  3867  3869  3873  3882  3884  3888  3890  3894
.MJMAP      142   158   163   179   196   200   217   221   360   382   393   400   406   410
            415   417   419   436   440   457   461   551   559   562   572   662   670   673
            683   808   819   832   843   848   857   864   866   886   890   907   911  1051
           1055  1063  1072  1081  1092  1103  1105  1115  1117  1240  1251  1261  1268  1270
           1280  1282  1459  1463  1469  1473  1483  1492  1503  1514  1516  1526  1528  1662
           1664  1667  1669  1879  1883  1891  1896  1898  1975  1979  1982  1984  2223  2234
           2238  2242  2247  2262  2273  2278  2283  2285  2302  2306  2323  2327  2552  2563
           2567  2571  2576  2591  2602  2607  2612  2614  2631  2635  2652  2656  2816  2818
           2820  2822  2824  2826  2828  2830  2845  2849  2854  2858  2872  2927  2931  2948
           2952  3114  3116  3118  3120  3122  3124  3126  3128  3145  3149  3154  3158  3172
           3227  3231  3248  3252  3414  3416  3418  3420  3422  3424  3426  3428  3430  3432
           3449  3453  3458  3462  3476  3539  3543  3560  3564  3734  3736  3738  3740  3742
           3744  3746  3748  3750  3752  3754  3756  3771  3775  3780  3784  3798  3869  3873
           3890  3894
.MJZ        869  1110  1275  1521  2213  2542  2867  2908  2910  3167  3208  3210  3471  3520
           3522  3793  3850  3852
.MLDCT      144   165   188   209   362   368   428   449   878   899  2215  2256  2294  2315
           2544  2585  2623  2644  2919  2940  3219  3240  3531  3552  3861  3882
.MLDLM      371  2878  2882  2886  2890  2894  2898  2902  2906  3178  3182  3186  3190  3194
           3198  3202  3206  3482  3486  3490  3494  3498  3502  3506  3510  3514  3518  3804
           3808  3812  3816  3820  3824  3828  3832  3836  3840  3844  3848
.MMGC       144   148   150   152   154   165   169   171   173   175   192   194   198   200
            213   215   219   221   360   371   376   378   387   389   415   417   419   432
            434   438   453   455   459   551   553   555   557   559   564   566   568   570
            572   662   664   666   668   670   675   677   679   681   683   802   804   813
            815   826   828   837   839   853   864   866   882   884   888   903   905   909
           1068  1070  1077  1079  1088  1094  1096  1103  1110  1115  1117  1247  1253  1255
           1266  1268  1275  1280  1282  1479  1481  1488  1490  1499  1505  1507  1514  1521
           1526  1528  1664  1669  1879  1885  1887  1889  1896  1898  1975  1977  1979  1982
           1984  2213  2247  2254  2264  2266  2273  2278  2283  2285  2298  2300  2304  2319
           2321  2325  2542  2576  2583  2593  2595  2602  2607  2612  2614  2627  2629  2633
           2648  2650  2654  2816  2818  2820  2822  2824  2826  2828  2830  2835  2837  2839
           2854  2860  2862  2867  2872  2874  2878  2882  2886  2890  2894  2898  2902  2906
           2923  2925  2929  2944  2946  2950  3114  3116  3118  3120  3122  3124  3126  3128
           3133  3135  3137  3139  3154  3160  3162  3167  3172  3174  3178  3182  3186  3190
           3194  3198  3202  3206  3223  3225  3229  3244  3246  3250  3414  3416  3418  3420
           3422  3424  3426  3428  3430  3432  3437  3439  3441  3443  3458  3464  3466  3471
           3476  3478  3482  3486  3490  3494  3498  3502  3506  3510  3514  3518  3535  3537
           3541  3556  3558  3562  3734  3736  3738  3740  3742  3744  3746  3748  3750  3752
           3754  3756  3761  3763  3765  3780  3786  3788  3793  3798  3800  3804  3808  3812
           3816  3820  3824  3828  3832  3836  3840  3844  3848  3865  3867  3871  3886  3888
           3892
.MOENA      148   150   169   171   192   200   213   221   371   376   415   417   419   432
            453   551   553   559   564   566   572   662   664   670   675   677   683   802
            813   826   837   882   903  1070  1079  1103  1253  [illegible]  1481  1490  1514  1664
           1669  1979  1984  2273  2278  2298  2319  2602  2607  [illegible]  2648  2837  2878  2882
           2886  2890  2894  2898  2902  2906  2923  2944  3135  3178  3182  3186  3190  3194
           3198  3202  3206  3223  3244  3439  3482  3486  3490  3494  3498  3502  3506  3510
           3514  3518  3535  3556  3763  3804  3808  3812  3816  3820  3824  3828  3832  3836
           3840  3844  3848  3865  3886
.MOR        144   146   150   154   165   167   171   175   192   198   200   213   219   221
```

```
           360   366   376   378   387   389   415   417   419   432   438   453   459   557
           570   668   681   802   804   806   813   815   826   828   839   841   882   888
           903   909  1079  1096  1103  1253  1257  1259  1268  1490  1507  1514  1887  1977
          1979  2230  2266  2273  2278  2298  2304  2319  2325  2559  2595  2602  2607  2627
          2633  2648  2654  2816  2818  2820  2822  2824  2826  2828  2830  2862  2923  2929
          2944  2950  3114  3116  3118  3120  3122  3124  3126  3128  3162  3223  3229  3244
          3250  3414  3416  3418  3420  3422  3424  3426  3428  3430  3432  3466  3535  3541
          3556  3562  3734  3736  3738  3740  3742  3744  3746  3748  3750  3752  3754  3756
          3788  3865  3871  3886  3892
.MPLUS     186   190   207   211   364   368   410   426   430   447   451   876   880   897
           901  1098  1242  1255  1509  1664  1669  2240  2292  2296  2313  2317  2569  2621
          2625  2642  2646  2917  2921  2938  2942  3217  3221  3238  3242  3529  3533  3550
          3554  3859  3863  3880  3884
.MRDLM     387  2816  2818  2820  2822  2824  2826  2828  2830  3114  3116  3118  3120  3122
          3124  3126  3128  3414  3416  3418  3420  3422  3424  3426  3428  3430  3432  3734
          3736  3738  3740  3742  3744  3746  3748  3750  3752  3754  3756
.MRMIN     156   177
.MRPCT     146   167   190   211   366   398   430   451   880   901  2217  2258  2296  2317
          2546  2587  2625  2646  2921  2942  3221  3242  3533  3554  3863  3884
.MSOA      146   148   150   167   169   171   192   198   200   213   219   221   362   371
           376   406   408   415   417   419   432   438   453   459   551   553   559   562
           564   566   572   662   664   670   673   675   677   683   800   802   806   813
           817   826   830   837   841   882   888   903   909  1070  1079  1103  1242  1253
          1268  1481  1490  1514  1662  1667  1979  1984  2230  2240  2273  2278  2298  2304
          2319  2325  2559  2569  2602  2607  2627  2633  2648  2654  2837  2841  2878  2882
          2886  2890  2894  2898  2902  2906  2923  2929  2944  2950  3135  3141  3178  3182
          3186  3190  3194  3198  3202  3206  3223  3229  3244  3250  3439  3445  3482  3486
          3490  3494  3498  3502  3506  3510  3514  3518  3535  3541  3556  3562  3763  3767
          3804  3808  3812  3816  3820  3824  3828  3832  3836  3840  3844  3848  3865  3871
          3886  3892
.MSOB      186   207   364   366   368   410   426   447   824   876   897  1051  1098  1240
          1255  1257  1259  1459  1509  1664  1669  2219  2292  2313  2548  2621  2642  2917
          2938  3217  3238  3529  3550  3859  3880
.MSAB      156   177   190   211   380   391   430   451   880   901  2296  2317  2625  2646
          2921  2942  3221  3242  3533  3554  3863  3884
.MSDO      144   152   154   165   173   175   184   205   360   378   387   389   424   445
           555   557   568   570   666   668   679   681   804   815   828   839   874   895
          1096  1507  1887  1977  2266  2290  2311  2595  2619  2640  2816  2818  2820  2822
          2824  2826  2828  2830  2862  2915  2936  3114  3116  3118  3120  3122  3124  3126
          3128  3137  3162  3215  3236  3414  3416  3418  3420  3422  3424  3426  3428  3430
          3432  3441  3466  3527  3548  3734  3736  3738  3740  3742  3744  3746  3748  3750
          3752  3754  3756  3788  3857  3878
.MSELC     813   837   853   864  1068  1077  1088  1094  1115  1247  1255  1266  1280  1479
          1488  1499  1505  1526  1879  1885  1889  1896  1975  1982  2247  2254  2264  2283
          2576  2583  2593  2612  2854  2860  2867  2872  3154  3160  3167  3172  3458  3464
          3471  3476  3765  3780  3786  3793  3798
.MSELE     192   194   198   200   213   215   219   221   415   417   419   432   434   438
           453   455   459   559   572   670   683   882   884   888   903   905   909  1070
          1079  1103  1268  1481  1490  1514  1664  1669  1979  1984  2273  2278  2298  2300
          2304  2319  2321  2325  2602  2607  2627  2629  2633  2648  2650  2654  2923  2925
          2929  2944  2946  2950  3223  3225  3229  3244  3246  3250  3535  3537  3541  3556
          3558  3562  3865  3867  3871  3886  3888  3892
.MSELF     148   169   376   387   551   564   662   675   802   826  1253  2837  2839  3139
```

```
         3443  3763
.MSELM    150   154   171   175   378   389   553   557   566   570   664   668   677   681       SEQ 1993
          804   815   828   839   866  1096  1110  1117  1275  1282  1507  1521  1528  1887
         1898  1977  2213  2266  2285  2542  2595  2614  2835  2862  2874  3133  3135  3162
         3174  3437  3439  3466  3478  3761  3788  3800
.MSELP    152   173   555   568   666   679  3137  3441
.MSKCN    144   165   360
.MSMIN    380   391
.MTIM    1667  1669
.MXNOR    148   169   371   553   564   566   664   675   677   817   824   830   837  1070
         1481
.RA       156   177   190   192   198   200   211   213   219   221   371   376   380   391
          415   417   419   430   432   438   451   453   459   559   572   670   683   802
          806   813   817   826   830   837   841   880   882   888   901   903   909  1070
         1079  1103  1242  1253  1268  1481  1490  1514  2230  2240  2273  2278  2296  2298
         2304  2317  2319  2325  2559  2569  2602  2607  2625  2627  2633  2646  2648  2654
         2837  2841  2878  2882  2886  2890  2894  2898  2902  2906  2921  2923  2929  2942
         2944  2950  3135  3141  3178  3182  3186  3190  3194  3198  3202  3206  3221  3223
         3229  3242  3244  3250  3439  3445  3482  3486  3490  3494  3498  3502  3506  3510
         3514  3518  3533  3535  3541  3554  3556  3562  3763  3767  3804  3808  3812  3816
         3820  3824  3828  3832  3836  3840  3844  3848  3863  3865  3871  3884  3886  3892
.RAND     184   205   362   406   408   424   445   551   562   662   673   800   874   895
         1051  1240  1459  1662  1667  1984  2219  2290  2311  2548  2619  2640  2915  2936
         3215  3236  3527  3548  3857  3878
.RB       148   154   156   169   175   177   184   186   190   205   207   211   360   362
          364   366   368   376   378   380   387   389   391   406   408   410   424   426
          430   445   447   451   557   570   668   681   800   804   815   824   828   839
          874   876   880   895   897   901  1051  1096  1098  1240  1242  1255  1257  1259
         1459  1507  1509  1887  2219  2240  2266  2290  2292  2296  2311  2313  2317  2548
         2569  2595  2619  2621  2625  2640  2642  2646  2816  2818  2820  2822  2824  2826
         2828  2830  2862  2915  2917  2921  2936  2938  2942  3114  3116  3118  3120  3122
         3124  3126  3128  3162  3215  3217  3221  3236  3238  3242  3414  3416  3418  3420
         3422  3424  3426  3428  3430  3432  3466  3527  3529  3533  3548  3550  3554  3734
         3736  3738  3740  3742  3744  3746  3748  3750  3752  3754  3756  3788  3857  3859
         3863  3878  3880  3884
.RCCCA    855  1090  1249  1501  1881  2260  2589  2856  3156  3460  3782
.RCCCP   1059  1467
.RCCFZ    156   177   380   391   806   817   830   841  1259
.RCCGC    194   215   434   455   884   905  2300  2321  2629  2650  2925  2946  3225  3246
         3537  3558  3867  3888
.RCCLW   1061  1471  2240  2569
.RCCMB   2230  2559  2841  3141  3445  3767
.RCCMP   2232  2236  2561  2565  2843  2847  3143  3147  3447  3451  3769  3773
.RCENA    156   177   194   215   380   391   434   455   806   817   830   841   855   884
          905  1059  1061  1090  1249  1259  1467  1471  1501  1881  2230  2232  2236  2240
         2260  2300  2321  2559  2561  2565  2569  2589  2629  2650  2841  2843  2847  2856
         2925  2946  3141  3143  3147  3156  3225  3246  3445  3447  3451  3460  3537  3558
         3767  3769  3773  3782  3867  3888
.RCJP     156   177   194   215   380   391   434   455   806   817   830   841   855   884
          905  1059  1061  1090  1249  1259  1467  1471  1501  1881  2230  2232  2236  2240
         2260  2300  2321  2559  2561  2565  2569  2589  2629  2650  2841  2843  2847  2856
         2925  2946  3141  3143  3147  3156  3225  3246  3445  3447  3451  3460  3537  3558
         3767  3769  3773  3782  3867  3888
```

1981

SEQ 1994

```
.RCJS      800   824  1053  1057  1461  1465  2221  2228  2550  2557  2876  2880  2884  2888
          2892  2896  2900  2904  3176  3180  3184  3188  3192  3196  3200  3204  3480  3484
          3488  3492  3496  3500  3504  3508  3512  3516  3802  3806  3810  3814  3818  3822
          3826  3830  3834  3838  3842  3846
.RCONT     148   150   152   154   169   171   173   175   184   186   192   198   205   207
           213   219   364   371   376   378   387   389   408   424   426   432   438   445
           447   453   459   553   555   557   564   566   568   570   664   666   668   675
           677   679   681   802   804   813   815   826   828   837   839   853   874   876
           882   888   895   897   903   909  1068  1070  1077  1079  1088  1094  1096  1242
          1247  1253  1255  1257  1266  1479  1481  1488  1490  1499  1505  1507  1885  1887
          1889  1977  2219  2254  2264  2266  2290  2292  2298  2304  2311  2313  2319  2325
          2548  2583  2593  2595  2619  2621  2627  2633  2640  2642  2648  2654  2835  2837
          2839  2860  2874  2878  2882  2886  2890  2894  2898  2902  2906  2915  2917  2923
          2929  2936  2938  2944  2950  3133  3135  3137  3139  3160  3174  3178  3182  3186
          3190  3194  3198  3202  3206  3215  3217  3223  3229  3236  3238  3244  3250  3437
          3439  3441  3443  3464  3478  3482  3486  3490  3494  3498  3502  3506  3510  3514
          3518  3527  3529  3535  3541  3548  3550  3556  3562  3761  3763  3765  3786  3800
          3804  3808  3812  3816  3820  3824  3828  3832  3836  3840  3844  3848  3857  3859
          3865  3871  3878  3880  3886  3892
.RCRTN     859  1098  1509  2268  2597  2862  3162  3466  3788
.RCRY      156   177   186   207   364   368   380   391   410   426   447   876   897  1098
          1242  1255  1509  1664  1669  2240  2292  2313  2569  2621  2642  2917  2938  3217
          3238  3529  3550  3859  3880
.RD        142   144   146   148   150   152   154   156   158   163   165   167   169   171
           173   175   177   179   184   186   190   192   194   196   198   200   205   207
           211   213   215   217   219   221   360   362   364   366   368   371   376   378
           382   387   389   393   398   400   406   408   410   415   417   419   424   426
           430   432   434   436   438   440   445   447   451   453   455   457   459   461
           551   553   555   557   559   562   564   566   568   570   572   662   664   666
           668   670   673   675   677   679   681   683   800   802   804   806   808   813
           815   817   819   824   826   828   830   832   837   839   841   843   848   853
           855   857   859   874   876   880   882   884   886   888   890   895   897   901
           903   905   907   909   911  1051  1053  1055  1057  1059  1061  1070  1079  1088
          1090  1092  1094  1096  1098  1103  1240  1242  1247  1249  1251  1253  1255  1257
          1259  1268  1459  1461  1463  1465  1467  1469  1471  1481  1490  1499  1501  1503
          1505  1507  1509  1514  1662  1664  1667  1669  1881  1883  1885  1887  1975  1977
          1979  1984  2219  2221  2228  2230  2232  2234  2236  2238  2240  2254  2256  2258
          2260  2262  2264  2266  2268  2273  2278  2290  2292  2296  2298  2300  2302  2304
          2306  2311  2313  2317  2319  2321  2323  2325  2327  2548  2550  2557  2559  2561
          2563  2565  2567  2569  2583  2585  2587  2589  2591  2593  2595  2597  2602  2607
          2619  2621  2625  2627  2629  2631  2633  2635  2640  2642  2646  2648  2650  2652
          2654  2656  2816  2818  2820  2822  2824  2826  2828  2830  2835  2837  2839  2841
          2843  2845  2847  2849  2854  2856  2858  2860  2862  2876  2878  2880  2882  2884
          2886  2888  2890  2892  2894  2896  2898  2900  2902  2904  2906  2915  2917  2921
          2923  2925  2927  2929  2931  2936  2938  2942  2944  2946  2948  2950  2952  3114
          3116  3118  3120  3122  3124  3126  3128  3133  3135  3137  3139  3141  3143  3145
          3147  3149  3154  3156  3158  3160  3162  3176  3178  3180  3182  3184  3186  3188
          3190  3192  3194  3196  3198  3200  3202  3204  3206  3215  3217  3221  3223  3225
          3227  3229  3231  3236  3238  3242  3244  3246  3248  3250  3252  3414  3416  3418
          3420  3422  3424  3426  3428  3430  3432  3437  3439  3441  3443  3445  3447  3449
          3451  3453  3458  3460  3462  3464  3466  3480  3482  3484  3486  3488  3490  3492
          3494  3496  3498  3500  3502  3504  3506  3508  3510  3512  3514  3516  3518  3527
          3529  3533  3535  3537  3539  3541  3543  3548  3550  3554  3556  3558  3560  3562
          3564  3734  3736  3738  3740  3742  3744  3746  3748  3750  3752  3754  3756  3761
          3763  3765  3767  3769  3771  3773  3775  3780  3782  3784  3786  3788  3802  3804
          3806  3808  3810  3812  3814  3816  3818  3820  3822  3824  3826  3828  3830  3832
          3834  3836  3838  3840  3842  3844  3846  3848  3857  3859  3863  3865  3867  3869
          3871  3873  3878  3880  3884  3886  3888  3890  3892  3894
.RJ        142   144   146   156   158   163   165   167   177   179   188   190   194   196
           200   209   211   215   217   221   360   362   366   368   380   382   391   393
           398   400   406   410   415   417   419   428   430   434   436   440   449   451
           455   457   461   551   559   562   572   662   670   673   683   800   806   808
           817   819   824   830   832   841   843   848   855   857   864   866   878   880
           884   886   890   899   901   905   907   911  1051  1053  1055  1057  1059  1061
          1063  1072  1081  1090  1092  1103  1105  1115  1117  1240  1249  1251  1259  1261
          1268  1270  1280  1282  1459  1461  1463  1465  1467  1469  1471  1473  1483  1492
          1501  1503  1514  1516  1526  1528  1662  1664  1667  1669  1879  1881  1883  1891
          1896  1898  1975  1979  1982  1984  2215  2217  2221  2223  2228  2230  2232  2234
          2236  2238  2240  2242  2247  2256  2258  2260  2262  2273  2278  2283  2285  2294
          2296  2300  2302  2306  2315  2317  2321  2323  2327  2544  2546  2550  2552  2557
          2559  2561  2563  2565  2567  2569  2571  2576  2585  2587  2589  2591  2602  2607
          2612  2614  2623  2625  2629  2631  2635  2644  2646  2650  2652  2656  2816  2818
          2820  2822  2824  2826  2828  2830  2841  2843  2845  2847  2849  2854  2856  2858
          2872  2876  2880  2884  2888  2892  2896  2900  2904  2919  2921  2925  2927  2931
          2940  2942  2946  2948  2952  3114  3116  3118  3120  3122  3124  3126  3128  3141
          3143  3145  3147  3149  3154  3156  3158  3172  3176  3180  3184  3188  3192  3196
          3200  3204  3219  3221  3225  3227  3231  3240  3242  3246  3248  3252  3414  3416
          3418  3420  3422  3424  3426  3428  3430  3432  3445  3447  3449  3451  3453  3458
          3460  3462  3476  3480  3484  3488  3492  3496  3500  3504  3508  3512  3516  3531
          3533  3537  3539  3543  3552  3554  3558  3560  3564  3734  3736  3738  3740  3742
          3744  3746  3748  3750  3752  3754  3756  3767  3769  3771  3773  3775  3780  3782
          3784  3798  3802  3806  3810  3814  3818  3822  3826  3830  3834  3838  3842  3846
          3861  3863  3867  3869  3873  3882  3884  3888  3890  3894
.RJMAP     142   158   163   179   196   200   217   221   360   382   393   400   406   410
           415   417   419   436   440   457   461   551   559   562   572   662   670   673
           683   808   819   832   843   848   857   864   866   886   890   907   911  1051
          1055  !063  1072  1081  1092  1103  1105  1115  1117  1240  1251  1261  1268  1270
          1280  1282  1459  1463  1469  1473  1483  1492  1503  1514  1516  1526  1528  1662
          1664  1667  1669  1879  1883  1891  1896  1898  1975  1979  1982  1984  2223  2234
          2238  2242  2247  2262  2273  2278  2283  2285  2302  2306  2323  2327  2552  2563
          2567  2571  2576  2591  2602  2607  2612  2614  2631  2635  2652  2656  2816  2818
          2820  2822  2824  2826  2828  2830  2845  2849  2854  2858  2872  2927  2931  2948
          2952  3114  3116  3118  3120  3122  3124  3126  3128  3145  3149  3154  3158  3172
          3227  3231  3248  3252  3414  3416  3418  3420  3422  3424  3426  3428  3430  3432
          3449  3453  3458  3462  3476  3539  3543  3560  3564  3734  3736  3738  3740  3742
          3744  3746  3748  3750  3752  3754  3756  3771  3775  3780  3784  3798  3869  3873
          3890  3894
.RJZ       869  1110  1275  1521  2213  2542  2867  2908  2910  3167  3208  3210  3471  3520
          3522  3793  3850  3852
.RLDCT     144   165   188   209   362   368   428   449   878   899  2215  2256  2294  2315
          2544  2585  2623  2644  2919  2940  3219  3240  3531  3552  3861  3882
.RLDLM     371  2878  2882  2886  2890  2894  2898  2902  2906  3178  3182  3186  3190  3194
          3198  3202  3206  3482  3486  3490  3494  3498  3502  3506  3510  3514  3518  3804
          3808  3812  3816  3820  3824  3828  3832  3836  3840  3844  3848
.RMGC      144   148   150   152   154   165   169   171   173   175   192   194   198   200
           213   215   219   221   360   371   376   378   387   389   415   417   419   432
          434   438   453   455   459   551   553   555   557   559   564   566   568   570
          572   662   664   666   668   670   675   677   679   681   683   802   804   813
          815   826   828   837   839   853   864   866   882   884   888   903   905   909
         1068  1070  1077  1079  1088  1094  1096  1103  1110  1115  1117  1247  1253  1255
         1266  1268  1275  1280  1282  1479  1481  1488  1490  1499  1505  1507  1514  1521
         1526  1528  1664  1669  1879  1885  1887  1889  1896  1898  1975  1977  1979  1982
         1984  2213  2247  2254  2264  2266  2273  2278  2283  2285  2298  2300  2304  2319
         2321  2325  2542  2576  2583  2593  2595  2602  2607  2612  2614  2627  2629  2633
         2648  2650  2654  2816  2818  2820  2822  2824  2826  2828  2830  2835  2837  2839
         2854  2860  2862  2867  2872  2874  2878  2882  2886  2890  2894  2898  2902  2906
         2923  2925  2929  2944  2946  2950  3114  3116  3118  3120  3122  3124  3126  3128
         3133  3135  3137  3139  3154  3160  3162  3167  3172  3174  3178  3182  3186  3190
         3194  3198  3202  3206  3223  3225  3229  3244  3246  3250  3414  3416  3418  3420
         3422  3424  3426  3428  3430  3432  3437  3439  3441  3443  3458  3464  3466  3471
         3476  3478  3482  3486  3490  3494  3498  3502  3506  3510  3514  3518  3535  3537
         3541  3556  3558  3562  3734  3736  3738  3740  3742  3744  3746  3748  3750  3752
         3754  3756  3761  3763  3765  3780  3786  3788  3793  3798  3800  3804  3808  3812
         3816  3820  3824  3828  3832  3836  3840  3844  3848  3865  3867  3871  3886  3888
         3892
.ROENA    148   150   169   171   192   200   213   221   371   376   415   417   419   432
          453   551   553   559   564   566   572   662   664   670   675   677   683   802
          813   826   837   882   903  1070  1079  1103  1253  1268  1481  1490  1514  1664
         1669  1979  1984  2273  2278  2298  2319  2602  2607  2627  2648  2837  2878  2882
         2886  2890  2894  2898  2902  2906  2923  2944  3135  3178  3182  3186  3190  3194
         3198  3202  3206  3223  3244  3439  3482  3486  3490  3494  3498  3502  3506  3510
         3514  3518  3535  3556  3763  3804  3808  3812  3816  3820  3824  3828  3832  3836
         3840  3844  3848  3865  3886
.ROR      144   146   150   154   165   167   171   175   192   198   200   213   219   221
          360   366   376   378   387   389   415   417   419   432   438   453   459   557
          570   668   681   802   804   806   813   815   826   828   839   841   882   888
          903   909  1079  1096  1103  1253  1257  1259  1268  1490  1507  1514  1887  1977
         1979  2230  2266  2273  2278  2298  2304  2319  2325  2559  2595  2602  2607  2627
         2633  2648  2654  2816  2818  2820  2822  2824  2826  2828  2830  2862  2923  2929
         2944  2950  3114  3116  3118  3120  3122  3124  3126  3128  3162  3223  3229  3244
         3250  3414  3416  3418  3420  3422  3424  3426  3428  3430  3432  3466  3535  3541
         3556  3562  3734  3736  3738  3740  3742  3744  3746  3748  3750  3752  3754  3756
         3788  3865  3871  3886  3892
.RPLUS    186   190   207   211   364   368   410   426   430   447   451   876   880   897
          901  1098  1242  1255  1509  1664  1669  2240  2292  2296  2313  2317  2569  2621
         2625  2642  2646  2917  2921  2938  2942  3217  3221  3238  3242  3529  3533  3550
         3554  3859  3863  3880  3884
.RRDLM    387  2816  2818  2820  2822  2824  2826  2828  2830  3114  3116  3118  3120  3122
         3124  3126  3128  3414  3416  3418  3420  3422  3424  3426  3428  3430  3432  3734
         3736  3738  3740  3742  3744  3746  3748  3750  3752  3754  3756
.RRMIN    156   177
.RRPCT    146   167   190   211   366   398   430   451   880   901  2217  2258  2296  2317
         2546  2587  2625  2646  2921  2942  3221  3242  3533  3554  3863  3884
.RSOA     146   148   150   167   169   171   192   198   200   213   219   221   362   371
          376   406   408   415   417   419   432   438   453   459   551   553   559   562
          564   566   572   662   664   670   673   675   677   683   800   802   806   813
          817   826   830   837   841   882   888   903   909  1070  1079  1103  1242  1253
         1268  1481  1490  1514  1662  1667  1979  1984  2230  2240  2273  2278  2298  2304
         2319  2325  2559  2569  2602  2607  2627  2633  2648  2654  2837  2841  2878  2882
```

```
            2886    2890    2894    2898    2902    2906    2923    2929    2944    2950    3135    3141    3178    3182
            3186    3190    3194    3198    3202    3206    3223    3229    3244    3250    3439    3445    3482    3486
            3490    3494    3498    3502    3506    3510    3514    3518    3535    3541    3556    3562    3763    3767
            3804    3808    3812    3816    3820    3824    3828    3832    3836    3840    3844    3848    3865    3871
            3886    3892
.RSOB        186     207     364     366     368     410     426     447     824     876     897    1051    1098    1240
            1255    1257    1259    1459    1509    1664    1669    2219    2292    2313    2548    2621    2642    2917
            2938    3217    3238    3529    3550    3859    3880
.RSAB        156     177     190     211     380     391     430     451     880     901    2296    2317    2625    2646
            2921    2942    3221    3242    3533    3554    3863    3884
.RSDO        144     152     154     165     173     175     184     205     360     378     387     389     424     445
             555     557     568     570     666     668     679     681     804     815     828     839     874     895
            1096    1507    1887    1977    2266    2290    2311    2595    2619    2640    2816    2818    2820    2822
            2824    2826    2828    2830    2862    2915    2936    3114    3116    3118    3120    3122    3124    3126
            3128    3137    3162    3215    3236    3414    3416    3418    3420    3422    3424    3426    3428    3430
            3432    3441    3466    3527    3548    3734    3736    3738    3740    3742    3744    3746    3748    3750
            3752    3754    3756    3788    3857    3878
.RSELC       813     837     853     864    1068    1077    1088    1094    1115    1247    1255    1266    1280    1479
            1488    1499    1505    1526    1879    1885    1889    1896    1975    1982    2247    2254    2264    2233
            2576    2583    2593    2612    2854    2860    2867    2872    3154    3160    3167    3172    3458    3464
            3471    3476    3765    3780    3786    3793    3798
.RSELE       192     194     198     200     213     215     219     221     415     417     419     432     434     438
             453     455     459     559     572     670     683     882     884     888     903     905     909    1070
            1079    1103    1268    1481    1490    1514    1664    1669    1979    1984    2273    2278    2298    2300
            2304    2319    2321    2325    2602    2607    2627    2629    2633    2648    2650    2654    2923    2925
            2929    2944    2946    2950    3223    3225    3229    3244    3246    3250    3535    3537    3541    3556
            3558    3562    3865    3867    3871    3886    3888    3892
.RSELF       148     169     376     387     551     564     662     675     802     826    1253    2837    2839    3139
            3443    3763
.RSELM       150     154     171     175     378     389     553     557     566     570     664     668     677     681
             804     815     828     839     866    1096    1110    1117    1275    1282    1507    1521    1528    1887
            1898    1977    2213    2266    2285    2542    2595    2614    2835    2862    2874    3133    3135    3162
            3174    3437    3439    3466    3478    3761    3788    3800
.RSELP       152     173     555     568     666     679    3137    3441
.RSKCN       144     165     360
.RSMIN       380     391
.RTIM       1667    1669
.RXNOR       148     169     371     553     564     566     664     675     677     817     824     830     837    1070
            1481
```

```
EQ 1985

ATABLE     538   540   542   649   651   653  1960  1962  1963  1965                          SEQ 1998
CALC       142   144   146   148   150   152   154   156   158   163   165   167   169   171
           173   175   177   179   184   186   188   190   192   194   196   198   200   205
           207   209   211   213   215   217   219   221   360   362   364   366   368   371
           376   378   380   382   387   389   391   393   398   400   406   408   410   415
           417   419   424   426   428   430   432   434   436   438   440   445   447   449
           451   453   455   457   459   461   551   553   555   557   559   562   564   566
           568   570   572   662   664   666   668   670   673   675   677   679   681   683
           800   802   804   806   808   813   815   817   819   824   826   828   830   832
           837   839   841   843   848   853   855   857   859   864   866   869   874   876
           878   880   882   884   886   888   890   895   897   899   901   903   905   907
           909   911  1051  1053  1055  1057  1059  1061  1063  1068  1070  1072  1077  1079
          1081  1088  1090  1092  1094  1096  1098  1103  1105  1110  1115  1117  1240  1242
          1247  1249  1251  1253  1255  1257  1259  1261  1266  1268  1270  1275  1280  1282
          1459  1461  1463  1465  1467  1469  1471  1473  1479  1481  1483  1488  1490  1492
          1499  1501  1503  1505  1507  1509  1514  1516  1521  1526  1528  1662  1664  1667
          1669  1879  1881  1883  1885  1887  1889  1891  1896  1898  1975  1977  1979  1982
          1984  2213  2215  2217  2219  2221  2223  2228  2230  2232  2234  2236  2238  2240
          2242  2247  2254  2256  2258  2260  2262  2264  2266  2268  2273  2278  2283  2285
          2290  2292  2294  2296  2298  2300  2302  2304  2306  2311  2313  2315  2317  2319
          2321  2323  2325  2327  2542  2544  2546  2548  2550  2552  2557  2559  2561  2563
          2565  2567  2569  2571  2576  2583  2585  2587  2589  2591  2593  2595  2597  2602
          2607  2612  2614  2619  2621  2623  2625  2627  2629  2631  2633  2635  2640  2642
          2644  2646  2648  2650  2652  2654  2656  2816  2818  2820  2822  2824  2826  2828
          2830  2835  2837  2839  2841  2843  2845  2847  2849  2854  2856  2858  2860  2862
          2867  2872  2874  2876  2878  2880  2882  2884  2886  2888  2890  2892  2894  2896
          2898  2900  2902  2904  2906  2908  2910  2915  2917  2919  2921  2923  2925  2927
          2929  2931  2936  2938  2940  2942  2944  2946  2948  2950  2952  3114  3116  3118
          3120  3122  3124  3126  3128  3133  3135  3137  3139  3141  3143  3145  3147  3149
          3154  3156  3158  3160  3162  3167  3172  3174  3176  3178  3180  3182  3184  3186
          3188  3190  3192  3194  3196  3198  3200  3202  3204  3206  3208  3210  3215  3217
          3219  3221  3223  3225  3227  3229  3231  3236  3238  3240  3242  3244  3246  3248
          3250  3252  3414  3416  3418  3420  3422  3424  3426  3428  3430  3432  3437  3439
          3441  3443  3445  3447  3449  3451  3453  3458  3460  3462  3464  3466  3471  3476
          3478  3480  3482  3484  3486  3488  3490  3492  3494  3496  3498  3500  3502  3504
          3506  3508  3510  3512  3514  3516  3518  3520  3522  3527  3529  3531  3533  3535
          3537  3539  3541  3543  3548  3550  3552  3554  3556  3558  3560  3562  3564  3734
          3736  3738  3740  3742  3744  3746  3748  3750  3752  3754  3756  3761  3763  3765
          3767  3769  3771  3773  3775  3780  3782  3784  3786  3788  3793  3798  3800  3802
          3804  3806  3808  3810  3812  3814  3816  3818  3820  3822  3824  3826  3828  3830
          3832  3834  3836  3838  3840  3842  3844  3846  3848  3850  3852  3857  3859  3861
          3863  3865  3867  3869  3871  3873  3878  3880  3882  3884  3886  3888  3890  3892
          3894
CONCAT     142   144   146   148   150   152   154   156   158   163   165   167   169   171
           173   175   177   179   184   186   188   190   192   194   196   198   200   205
           207   209   211   213   215   217   219   221   360   362   364   366   368   371
           376   378   380   382   387   389   391   393   398   400   406   408   410   415
           417   419   424   426   428   430   432   434   436   438   440   445   447   449
           451   453   455   457   459   461   551   553   555   557   559   562   564   566
           568   570   572   662   664   666   668   670   673   675   677   679   681   683
           800   802   804   806   808   813   815   817   819   824   826   828   830   832
           837   839   841   843   848   853   855   857   859   864   866   869   874   876
           878   880   882   884   886   888   890   895   897   899   901   903   905   907
```

```
              909    911   1051   1053   1055   1057   1059   1061   1063   1068   1070   1072   1077   1079
EQ 1986      1081   1088   1090   1092   1094   1096   1098   1103   1105   1110   1115   1117   1240   1242      SEQ 1999
             1247   1249   1251   1253   1255   1257   1259   1261   1266   1268   1270   1275   1280   1282
             1459   1461   1463   1465   1467   1469   1471   1473   1479   1481   1483   1488   1490   1492
             1499   1501   1503   1505   1507   1509   1514   1516   1521   1526   1528   1662   1664   1667
             1669   1879   1881   1883   1885   1887   1889   1891   1896   1898   1975   1977   1979   1982
             1984   2213   2215   2217   2219   2221   2223   2228   2230   2232   2234   2236   2238   2240
             2242   2247   2254   2256   2258   2260   2262   2264   2266   2268   2273   2278   2283   2285
             2290   2292   2294   2296   2298   2300   2302   2304   2306   2311   2313   2315   2317   2319
             2321   2323   2325   2327   2542   2544   2546   2548   2550   2552   2557   2559   2561   2563
             2565   2567   2569   2571   2576   2583   2585   2587   2589   2591   2593   2595   2597   2602
             2607   2612   2614   2619   2621   2623   2625   2627   2629   2631   2633   2635   2640   2642
             2644   2646   2648   2650   2652   2654   2656   2816   2818   2820   2822   2824   2826   2828
             2830   2835   2837   2839   2841   2843   2845   2847   2849   2854   2856   2858   2860   2862
             2867   2872   2874   2876   2878   2880   2882   2884   2886   2888   2890   2892   2894   2896
             2898   2900   2902   2904   2906   2908   2910   2915   2917   2919   2921   2923   2925   2927
             2929   2931   2936   2938   2940   2942   2944   2946   2948   2950   2952   3114   3116   3118
             3120   3122   3124   3126   3128   3133   3135   3137   3139   3141   3143   3145   3147   3149
             3154   3156   3158   3160   3162   3167   3172   3174   3176   3178   3180   3182   3184   3186
             3188   3190   3192   3194   3196   3198   3200   3202   3204   3206   3208   3210   3215   3217
             3219   3221   3223   3225   3227   3229   3231   3236   3238   3240   3242   3244   3246   3248
             3250   3252   3414   3416   3418   3420   3422   3424   3426   3428   3430   3432   3437   3439
             3441   3443   3445   3447   3449   3451   3453   3458   3460   3462   3464   3466   3471   3476
             3478   3480   3482   3484   3486   3488   3490   3492   3494   3496   3498   3500   3502   3504
             3506   3508   3510   3512   3514   3516   3518   3520   3522   3527   3529   3531   3533   3535
             3537   3539   3541   3543   3548   3550   3552   3554   3556   3558   3560   3562   3564   3734
             3736   3738   3740   3742   3744   3746   3748   3750   3752   3754   3756   3761   3763   3765
             3767   3769   3771   3773   3775   3780   3782   3784   3786   3788   3793   3798   3800   3802
             3804   3806   3808   3810   3812   3814   3816   3818   3820   3822   3824   3826   3828   3830
             3832   3834   3836   3838   3840   3842   3844   3846   3848   3850   3852   3857   3859   3861
             3863   3865   3867   3869   3871   3873   3878   3880   3882   3884   3886   3888   3890   3892
             3894
CTABLE        988    989    990    991    992   1188   1189   1190   1191   1192   1375   1376   1377   1378
             1379   1380
FIELD         142    144    146    148    150    152    154    156    158    163    165    167    169    171
              173    175    177    179    184    186    188    190    192    194    196    198    200    205
              207    209    211    213    215    217    219    221    360    362    364    366    368    371
              376    378    380    382    387    389    391    393    398    400    406    408    410    415
              417    419    424    426    428    430    432    434    436    438    440    445    447    449
              451    453    455    457    459    461    551    553    555    557    559    562    564    566
              568    570    572    662    664    666    668    670    673    675    677    679    681    683
              800    802    804    806    808    813    815    817    819    824    826    828    830    832
              837    839    841    843    848    853    855    857    859    864    866    869    874    876
              878    880    882    884    886    888    890    895    897    899    901    903    905    907
              909    911   1051   1053   1055   1057   1059   1061   1063   1068   1070   1072   1077   1079
             1081   1088   1090   1092   1094   1096   1098   1103   1105   1110   1115   1117   1240   1242
             1247   1249   1251   1253   1255   1257   1259   1261   1266   1268   1270   1275   1280   1282
             1459   1461   1463   1465   1467   1469   1471   1473   1479   1481   1483   1488   1490   1492
             1499   1501   1503   1505   1507   1509   1514   1516   1521   1526   1528   1662   1664   1667
             1669   1879   1881   1883   1885   1887   1889   1891   1896   1898   1975   1977   1979   1982
             1984   2213   2215   2217   2219   2221   2223   2228   2230   2232   2234   2236   2238   2240
             2242   2247   2254   2256   2258   2260   2262   2264   2266   2268   2273   2278   2283   2285
             2290   2292   2294   2296   2298   2300   2302   2304   2306   2311   2313   2315   2317   2319
```

```
SEQ 1987     2321   2323   2325   2327   2542   2544   2546   2548   2550   2552   2557   2559   2561   2563    SEQ 2000
             2565   2567   2569   2571   2576   2583   2585   2587   2589   2591   2593   2595   2597   2602
             2607   2612   2614   2619   2621   2623   2625   2627   2629   2631   2633   2635   2640   2642
             2644   2646   2648   2650   2652   2654   2656   2816   2818   2820   2822   2824   2826   2828
             2830   2835   2837   2839   2841   2843   2845   2847   2849   2854   2856   2858   2860   2862
             2867   2872   2874   2876   2878   2880   2882   2884   2886   2888   2890   2892   2894   2896
             2898   2900   2902   2904   2906   2908   2910   2915   2917   2919   2921   2923   2925   2927
             2929   2931   2936   2938   2940   2942   2944   2946   2948   2950   2952   3114   3116   3118
             3120   3122   3124   3126   3128   3133   3135   3137   3139   3141   3143   3145   3147   3149
             3154   3156   3158   3160   3162   3167   3172   3174   3176   3178   3180   3182   3184   3186
             3188   3190   3192   3194   3196   3198   3200   3202   3204   3206   3208   3210   3215   3217
             3219   3221   3223   3225   3227   3229   3231   3236   3238   3240   3242   3244   3246   3248
             3250   3252   3414   3416   3418   3420   3422   3424   3426   3428   3430   3432   3437   3439
             3441   3443   3445   3447   3449   3451   3453   3458   3460   3462   3464   3466   3471   3476
             3478   3480   3482   3484   3486   3488   3490   3492   3494   3496   3498   3500   3502   3504
             3506   3508   3510   3512   3514   3516   3518   3520   3522   3527   3529   3531   3533   3535
             3537   3539   3541   3543   3548   3550   3552   3554   3556   3558   3560   3562   3564   3734
             3736   3738   3740   3742   3744   3746   3748   3750   3752   3754   3756   3761   3763   3765
             3767   3769   3771   3773   3775   3780   3782   3784   3786   3788   3793   3798   3800   3802
             3804   3806   3808   3810   3812   3814   3816   3818   3820   3822   3824   3826   3828   3830
             3832   3834   3836   3838   3840   3842   3844   3846   3848   3850   3852   3857   3859   3861
             3863   3865   3867   3869   3871   3873   3878   3880   3882   3884   3886   3888   3890   3892
             3894
GO             80     82     88     94     95    270    272    278    284    285    314    317    319    320
              324    327    329    330    334    337    339    340    506    508    514    520    617    619
              625    631    734    736    748    749    777    779    780    784    788    956    958    969
              970    996   1031   1034   1036   1038   1039   1042   1156   1158   1169   1170 11[illegible] 1226
             1233   1336   1338   1349   1350   1404   1439   1442   1444   1446   1447   1450   1573   1579
             1581   1588   1592   1622   1624   1626   1628   1630   1631   1702   1704   1717   1718   1746
             1748   1750   1754   1758   1764   1765   1817   1823   1928   1930   1936   1942   2030   2032
             2038   2044   2053   2055   2056   2063   2065   2066   2091   2094   2096   2098   2100   2102
             2106   2372   2374   2380   2386   2395   2397   2398   2405   2407   2408   2433   2436   2438
             2440   2442   2444   2448   2701   2703   2709   2715   2746   2748   2750   2754   2758   2761
             2763   2765   2767   2999   3001   3007   3013   3044   3046   3048   3052   3056   3059   3061
             3063   3065   3297   3299   3305   3311   3342   3344   3346   3350   3354   3357   3359   3361
             3363   3606   3608   3614   3620   3651   3653   3655   3659   3663   3666   3668   3670   3672
             3679   3682   3684   3686
MFLD         142#    142   144#    144   146#    146   148#    148   150#    150   152#    152   154#    154
             156#    156   158#    158   163#    163   165#    165   167#    167   169#    169   171#    171
             173#    173   175#    175   177#    177   179#    179   184#    184   186#    186   188#    188
             190#    190   192#    192   194#    194   196#    196   198#    198   200#    200   205#    205
             207#    207   209#    209   211#    211   213#    213   215#    215   217#    217   219#    219
             221#    221   360#    360   362#    362   364#    364   366#    366   368#    368   371#    371
             376#    376   378#    378   380#    380   382#    382   387#    387   389#    389   391#    391
             393#    393   398#    398   400#    400   406#    406   408#    408   410#    410   415#    415
             417#    417   419#    419   424#    424   426#    426   428#    428   430#    430   432#    432
             434#    434   436#    436   438#    438   440#    440   445#    445   447#    447   449#    449
             451#    451   453#    453   455#    455   457#    457   459#    459   461#    461   551#    551
             553#    553   555#    555   557#    557   559#    559   562#    562   564#    564   566#    566
             568#    568   570#    570   572#    572   662#    662   664#    664   666#    666   668#    668
             670#    670   673#    673   675#    675   677#    677   679#    679   681#    681   683#    683
             800#    800   802#    802   804#    804   806#    806   808#    808   813#    813   815#    815
             817#    817   819#    819   824#    824   826#    826   828#    828   830#    830   832#    832
```

SEQ 1988

837#	837	839#	839	841#	841	843#	843	848#	848	853#	853	855#	855
857#	857	859#	859	864#	864	866#	866	869#	869	874#	874	876#	876
878#	878	880#	880	882#	882	884#	884	886#	886	888#	888	890#	890
895#	895	897#	897	899#	899	901#	901	903#	903	905#	905	907#	907
909#	909	911#	911	1051#	1051	1053#	1053	1055#	1055	1057#	1057	1059#	1059
1061#	1061	1063#	1063	1068#	1068	1070#	1070	1072#	1072	1077#	1077	1079#	1079
1081#	1081	1088#	1088	1090#	1090	1092#	1092	1094#	1094	1096#	1096	1098#	1098
1103#	1103	1105#	1105	1110#	1110	1115#	1115	1117#	1117	1240#	1240	1242#	1242
1247#	1247	1249#	1249	1251#	1251	1253#	1253	1255#	1255	1257#	1257	1259#	1259
1261#	1261	1266#	1266	1268#	1268	1270#	1270	1275#	1275	1280#	1280	1282#	1282
1459#	1459	1461#	1461	1463#	1463	1465#	1465	1467#	1467	1469#	1469	1471#	1471
1473#	1473	1479#	1479	1481#	1481	1483#	1483	1488#	1488	1490#	1490	1492#	1492
1499#	1499	1501#	1501	1503#	1503	1505#	1505	1507#	1507	1509#	1509	1514#	1514
1516#	1516	1521#	1521	1526#	1526	1528#	1528	1662#	1662	1664#	1664	1667#	1667
1669#	1669	1879#	1879	1881#	1881	1883#	1883	1885#	1885	1887#	1887	1889#	1889
1891#	1891	1896#	1896	1898#	1898	1975#	1975	1977#	1977	1979#	1979	1982#	1982
1984#	1984	2213#	2213	2215#	2215	2217#	2217	2219#	2219	2221#	2221	2223#	2223
2228#	2228	2230#	2230	2232#	2232	2234#	2234	2236#	2236	2238#	2238	2240#	2240
2242#	2242	2247#	2247	2254#	2254	2256#	2256	2258#	2258	2260#	2260	2262#	2262
2264#	2264	2266#	2266	2268#	2268	2273#	2273	2278#	2278	2283#	2283	2285#	2285
2290#	2290	2292#	2292	2294#	2294	2296#	2296	2298#	2298	2300#	2300	2302#	2302
2304#	2304	2306#	2306	2311#	2311	2313#	2313	2315#	2315	2317#	2317	2319#	2319
2321#	2321	2323#	2323	2325#	2325	2327#	2327	2542#	2542	2544#	2544	2546#	2546
2548#	2548	2550#	2550	2552#	2552	2557#	2557	2559#	2559	2561#	2561	2563#	2563
2565#	2565	2567#	2567	2569#	2569	2571#	2571	2576#	2576	2583#	2583	2585#	2585
2587#	2587	2589#	2589	2591#	2591	2593#	2593	2595#	2595	2597#	2597	2602#	2602
2607#	2607	2612#	2612	2614#	2614	2619#	2619	2621#	2621	2623#	2623	2625#	2625
2627#	2627	2629#	2629	2631#	2631	2633#	2633	2635#	2635	2640#	2640	2642#	2642
2644#	2644	2646#	2646	2648#	2648	2650#	2650	2652#	2652	2654#	2654	2656#	2656
2816#	2816	2818#	2818	2820#	2820	2822#	2822	2824#	2824	2826#	2826	2828#	2828
2830#	2830	2835#	2835	2837#	2837	2839#	2839	2841#	2841	2843#	2843	2845#	2845
2847#	2847	2849#	2849	2854#	2854	2856#	2856	2858#	2858	2860#	2860	2862#	2862
2867#	2867	2872#	2872	2874#	2874	2876#	2876	2878#	2878	2880#	2880	2882#	2882
2884#	2884	2886#	2886	2888#	2888	2890#	2890	2892#	2892	2894#	2894	2896#	2896
2898#	2898	2900#	2900	2902#	2902	2904#	2904	2906#	2906	2908#	2908	2910#	2910
2915#	2915	2917#	2917	2919#	2919	2921#	2921	2923#	2923	2925#	2925	2927#	2927
2929#	2929	2931#	2931	2936#	2936	2938#	2938	2940#	2940	2942#	2942	2944#	2944
2946#	2946	2948#	2948	2950#	2950	2952#	2952	3114#	3114	3116#	3116	3118#	3118
3120#	3120	3122#	3122	3124#	3124	3126#	3126	3128#	3128	3133#	3133	3135#	3135
3137#	3137	3139#	3139	3141#	3141	3143#	3143	3145#	3145	3147#	3147	3149#	3149
3154#	3154	3156#	3156	3158#	3158	3160#	3160	3162#	3162	3167#	3167	3172#	3172
3174#	3174	3176#	3176	3178#	3178	3180#	3180	3182#	3182	3184#	3184	3186#	3186
3188#	3188	3190#	3190	3192#	3192	3194#	3194	3196#	3196	3198#	3198	3200#	3200
3202#	3202	3204#	3204	3206#	3206	3208#	3208	3210#	3210	3215#	3215	3217#	3217
3219#	3219	3221#	3221	3223#	3223	3225#	3225	3227#	3227	3229#	3229	3231#	3231
3236#	3236	3238#	3238	3240#	3240	3242#	3242	3244#	3244	3246#	3246	3248#	3248
3250#	3250	3252#	3252	3414#	3414	3416#	3416	3418#	3418	3420#	3420	3422#	3422
3424#	3424	3426#	3426	3428#	3428	3430#	3430	3432#	3432	3437#	3437	3439#	3439
3441#	3441	3443#	3443	3445#	3445	3447#	3447	3449#	3449	3451#	3451	3453#	3453
3458#	3458	3460#	3460	3462#	3462	3464#	3464	3466#	3466	3471#	3471	3476#	3476
3478#	3478	3480#	3480	3482#	3482	3484#	3484	3486#	3486	3488#	3488	3490#	3490
3492#	3492	3494#	3494	3496#	3496	3498#	3498	3500#	3500	3502#	3502	3504#	3504
3506#	3506	3508#	3508	3510#	3510	3512#	3512	3514#	3514	3516#	3516	3518#	3518

SEQ 2001

```
SEQ 1989  3520#  3520   3522#  3522   3527#  3527   3529#  3529   3531#  3531   3533#  3533   3535#  3535   SEQ 2002
          3537#  3537   3539#  3539   3541#  3541   3543#  3543   3548#  3548   3550#  3550   3552#  3552
          3554#  3554   3556#  3556   3558#  3558   3560#  3560   3562#  3562   3564#  3564   3734#  3734
          3736#  3736   3738#  3738   3740#  3740   3742#  3742   3744#  3744   3746#  3746   3748#  3748
          3750#  3750   3752#  3752   3754#  3754   3756#  3756   3761#  3761   3763#  3763   3765#  3765
          3767#  3767   3769#  3769   3771#  3771   3773#  3773   3775#  3775   3780#  3780   3782#  3782
          3784#  3784   3786#  3786   3788#  3788   3793#  3793   3798#  3798   3800#  3800   3802#  3802
          3804#  3804   3806#  3806   3808#  3808   3810#  3810   3812#  3812   3814#  3814   3816#  3816
          3818#  3818   3820#  3820   3822#  3822   3824#  3824   3826#  3826   3828#  3828   3830#  3830
          3832#  3832   3834#  3834   3836#  3836   3838#  3838   3840#  3840   3842#  3842   3844#  3844
          3846#  3846   3848#  3848   3850#  3850   3852#  3852   3857#  3857   3859#  3859   3861#  3861
          3863#  3863   3865#  3865   3867#  3867   3869#  3869   3871#  3871   3873#  3873   3878#  3878
          3880#  3880   3882#  3882   3884#  3884   3886#  3886   3888#  3888   3890#  3890   3892#  3892
          3894#  3894
MWORD      142    144    146    148    150    152    154    156    158    163    165    167    169    171
           173    175    177    179    184    186    188    190    192    194    196    198    200    205
           207    209    211    213    215    217    219    221    360    362    364    366    368    371
           376    378    380    382    387    389    391    393    398    400    406    408    410    415
           417    419    424    426    428    430    432    434    436    438    440    445    447    449
           451    453    455    457    459    461    551    553    555    557    559    562    564    566
           568    570    572    662    664    666    668    670    673    675    677    679    681    683
           800    802    804    806    808    813    815    817    819    824    826    828    830    832
           837    839    841    843    848    853    855    857    859    864    866    869    874    876
           878    880    882    884    886    888    890    895    897    899    901    903    905    907
           909    911   1051   1053   1055   1057   1059   1061   1063   1068   1070   1072   1077   1079
          1081   1088   1090   1092   1094   1096   1098   1103   1105   1110   1115   1117   1240   1242
          1247   1249   1251   1253   1255   1257   1259   1261   1266   1268   1270   1275   1280   1282
          1459   1461   1463   1465   1467   1469   1471   1473   1479   1481   1483   1488   1490   1492
          1499   1501   1503   1505   1507   1509   1514   1516   1521   1526   1528   1662   1664   1667
          1669   1879   1881   1883   1885   1887   1889   1891   1896   1898   1975   1977   1979   1982
          1984   2213   2215   2217   2219   2221   2223   2228   2230   2232   2234   2236   2238   2240
          2242   2247   2254   2256   2258   2260   2262   2264   2266   2268   2273   2278   2283   2285
          2290   2292   2294   2296   2298   2300   2302   2304   2306   2311   2313   2315   2317   2319
          2321   2323   2325   2327   2542   2544   2546   2548   2550   2552   2557   2559   2561   2563
          2565   2567   2569   2571   2576   2583   2585   2587   2589   2591   2593   2595   2597   2602
          2607   2612   2614   2619   2621   2623   2625   2627   2629   2631   2633   2635   2640   2642
          2644   2646   2648   2650   2652   2654   2656   2816   2818   2820   2822   2824   2826   2828
          2830   2835   2837   2839   2841   2843   2845   2847   2849   2854   2856   2858   2860   2862
          2867   2872   2874   2876   2878   2880   2882   2884   2886   2888   2890   2892   2894   2896
          2898   2900   2902   2904   2906   2908   2910   2915   2917   2919   2921   2923   2925   2927
          2929   2931   2936   2938   2940   2942   2944   2946   2948   2950   2952   3114   3116   3118
          3120   3122   3124   3126   3128   3133   3135   3137   3139   3141   3143   3145   3147   3149
          3154   3156   3158   3160   3162   3167   3172   3174   3176   3178   3180   3182   3184   3186
          3188   3190   3192   3194   3196   3198   3200   3202   3204   3206   3208   3210   3215   3217
          3219   3221   3223   3225   3227   3229   3231   3236   3238   3240   3242   3244   3246   3248
          3250   3252   3414   3416   3418   3420   3422   3424   3426   3428   3430   3432   3437   3439
          3441   3443   3445   3447   3449   3451   3453   3458   3460   3462   3464   3466   3471   3476
          3478   3480   3482   3484   3486   3488   3490   3492   3494   3496   3498   3500   3502   3504
          3506   3508   3510   3512   3514   3516   3518   3520   3522   3527   3529   3531   3533   3535
          3537   3539   3541   3543   3548   3550   3552   3554   3556   3558   3560   3562   3564   3734
          3736   3738   3740   3742   3744   3746   3748   3750   3752   3754   3756   3761   3763   3765
          3767   3769   3771   3773   3775   3780   3782   3784   3786   3788   3793   3798   3800   3802
          3804   3806   3808   3810   3812   3814   3816   3818   3820   3822   3824   3826   3828   3830
          3832   3834   3836   3838   3840   3842   3844   3846   3848   3850   3852   3857   3859   3861       SEQ 2003
          3863   3865   3867   3869   3871   3873   3878   3880   3882   3884   3886   3888   3890   3892
          3894
PDEC      1020   1219   1428
PNT3      3102   3402
PNTDEC    1025   1433   1644   1648   1652   1656   1871
PNTHW      349    352   1046   1454   1832   1835   1838   1843   2201   2530   2804
PNTMSG     135    136    345    346    347    350   1017   1018   1019   1021   1023   1026   1044   1216
          1217   1218   1220   1425   1426   1427   1429   1431   1434   1452   1638   1642   1645   1646
          1649   1650   1653   1654   1657   1813   1815   1819   1821   1830   1833   1836   1841   1844
          1847   1849   1851   1853   1855   1857   1859   1861   1863   1864   1867   1868   1872   2199
          2204   2206   2528   2533   2535   2802   2807   2809   3100   3105   3107   3400   3405   3407
          3720   3725   3727
PNTOCS    1866   1874   3722
RTN         83    109    133    273    299    344    353    509    534    620    645    737    763    789
           959    984   1002   1043   1047   1159   1184   1202   1230   1234   1365   1371   1396   1410
          1451   1455   1577   1582   1618   1632   1641   1658   1705   1732   1759   1796   1824   1839
          1875   1931   1956   2033   2078   2110   2207   2375   2420   2452   2536   2704   2729   2768
          2773   2782   2810   3002   3027   3066   3071   3080   3108   3300   3325   3364   3369   3378
          3408   3609   3634   3673   3687   3696   3728
SCOPER     102    292    527    638    756    977   1177   1357   1612   1725   1949   2071   2413   2722
          3020   3318   3627
TMSG      1021   1026   1220   1429   1434   1645   1649   1653   1657   1847   1849   1851   1853   1855
          1857   1859   1861   1863   1864   1868   1872
TMSGC      135    136    345    346    347    350   1017   1018   1019   1023   1044   1216   1217   1218
          1425   1426   1427   1431   1452   1638   1642   1646   1650   1654   1813   1815   1819   1821
          1830   1833   1836   1841   1844   1867   2199   2204   2206   2528   2533   2535   2802   2807
          2809   3100   3105   3107   3400   3405   3407   3720   3725   3727
TTABLE     115    116    117    121    122    123    124    303    304    305    306    767    768    769
           770    771   1736   1737   1738   1739   1740   2082   2083   2084   2085   2424   2425   2426
          2427   2733   2734   2735   2736   2737   2738   2739   2740   3031   3032   3033   3034   3035
          3036   3037   3038   3329   3330   3331   3332   3333   3334   3335   3336   3638   3639   3640
          3641   3642   3643   3644   3645
```

```
DFPTAI  MACRO %53B(1242) 10:15  1-Jul-85 Page 1                                                          SEQ 2004
```

```
                TITLE   DFPTAI
                SUBTTL  Port Controller Utility Package

                SALL

                SEARCH  DFPTA,MONSYM,MACSYM

; Register Read/write routines

                ENTRY   .CONI,.CONO,.DATAO,.DATAI
                ENTRY   LDEBUF,LDRAR,LDCRAM,LDCSR
                ENTRY   RDEBUF,RDLAR,RDCRAM,RDCSR

; Port specific utility routines

                ENTRY   SETEBU,SETLAR,CLREBU,CHKCSR
                ENTRY   IPACLR,IPASRT,IPASTP,ERESET,IPASST,ISTOP
                INTERN  SNEXT,SLAST,EADDR,AADDR,SDATA,HAVENI,HAVECI

; PI system handling routines

                ENTRY   INITPI,INITPD,.PIOFF,.PION,PISYOF
                ENTRY   PISYON,SETVEC,.INPNT
                INTERN  INTNUM,INTPC,INTTYP,INTCON,INTAPR,INTTIM,INTUSE
                INTERN  INTCSR,INTEND,.INWD1,.INWD2,IDELAY

; Channel handling routines

                ENTRY   CHINIT,GENCCW
                ENTRY   GETLOG,CCWPNT,LOGPNT,CHDATA
                INTERN  CBASE,LOGBUF

; User mode support routines

                ENTRY   DSETUP,DEVREQ,DEVREL,REQ1,REQ2,REL
                ENTRY   USRION,USRIOF

; Miscellaneous routines

                ENTRY   FFF                             ; fatal error routine
                ENTRY   PFSET,PFCLR                     ; page fail handling routines
                INTERN  PFAIL,PFAPR,PF500,PF501,PF502   ; page fail data

; EXTERN's located in DFPTAU.MAC (Utility Routines module)

                EXTERN  .CLOSE,.EXIT,SWITT,RUNTME,DEBTIM,USTIM
```

```
                                        ; EXTERN's located in DFPTA1.MAC (Mainline code..)

                                                EXTERN  UDEBUG,CBUF,CSTWRD,PORTNI,PORTCI,NISEL,CISEL
                                                EXTERN  PMODE,ODELAY


                                        ;#**********************************************************************
                                        ; ZI - Address for use in DDT
                                        ;#**********************************************************************

        000000'                         ZI:                             ; address of 00000'

```

```
                        SUBTTL  Port Register Read/Write Routines

;#*************************************************************************
;* Notes On I/O Routine Design
;
; 1.  All code in this file is re-entrant  with the exception of the
;        user mode stuff which need not be since the PI system cannot
;        be used in user mode.
;
; 2.  No AC's are modified unless they return data.
;
; 3.   If in (exec mode) with a port it is necessary for the  channel
;        routines to do a limited amount of paging, a location "CBASE"
;        must be set up as a  paged address of "0".  Then the routines
;        will then work properly.
;
; The following routines are given:
;
; Port Register Load/Read Routines:
;
;       LDEBUF          This is a DATAO with 'Diag Test EBUF' bit set
;                       in the CSR Register.
;
;       LDRAR           This is a DATAO with 'Diag Test EBUF' bit not
;                       set in the CSR Register, and Bit 0 set in the
;                       data given.  The address to put in the RAR is
;                       in bits 1-13.
;
;       LDCRAM          This is a DATAO with 'Diag Test EBUF' bit not
;                       set in the CSR Register, and Bit 0 zero in the
;                       data given.  The data to  load  in  CRAM is in
;                       bits 6-35.
;
;       LDCSR           This is simply a CONO.
;
;       RDEBUF          This is a DATAI with 'Diag Test EBUF' bit set
;                       in the CSR Register.
;
;       RDLAR           This is a DATAI with 'Diag Test EBUF' bit not
;                       set and 'Diag Sel LAR'  set  in  CSR Register
;                       Register.  The data returned is in bits 1-12.
;                       Bit 13 is returned as the  corresponding  bit
;                       of the RAR.
;
;       RDCRAM          This is a DATAI with 'Diag Test EBUF' bit not
;                       set and 'Diag Sel LAR' not set in  CSR Regis-
;                       ter.  The data returned is in bits 6-35.
;
;       RDCSR           This is simply a CONI.
;
;#*************************************************************************
```

```
                                        ;#***********************************************************
                                        ;* Port Register Load Routines
                                        ;
                                        ; Port registers are loaded by a CONO or DATAO.  The data to be
                                        ; written  is given right justified in AC1.  Each routine  will
                                        ; will position the data correctly.
                                        ;
                                        ; Example:  MOVE  AC1,FOO                ; data to write to register
                                        ;           GO    LDXXXX                 ; XXXX (Register description)
                                        ;           Return                       ; no errors are possible
                                        ;#***********************************************************

                                        ;#***********************************************************
                                        ;* LDEBUF - Load EBUF
                                        ;
                                        ; This routine assumes that 'Diag Test EBUF' bit is already set
                                        ; in the CSR Register, so a simple DATAO will load the register.
                                        ;#***********************************************************

000000' 332 00 0 00 000000*             LDEBUF: SKIPE   UDEBUG                  ; debug mode?
000001' 263 17 0 00 000000                      RTN                             ; yes - exit
000002' 261 17 0 00 000002                      PUT     2                       ; save AC2
000003' 200 02 0 00 003170'                     MOVE    2,[DATAO 0,1]           ; build a DATAO
000004' 434 02 0 00 000016                      IOR     2,MBCN                  ; include the device code
000005' 256 00 0 00 000002                      XCT     2                       ; do the DATAO
000006' 262 17 0 00 000002                      GET     2                       ; restore AC2
000007' 263 17 0 00 000000                      RTN


                                        ;#***********************************************************
                                        ;* LDCSR - Load CSR (This is simply a CONO).
                                        ;#***********************************************************

000010' 332 00 0 00 000000*             LDCSR:  SKIPE   UDEBUG                  ; debug mode?
000011' 263 17 0 00 000000                      RTN                             ; yes - exit
000012' 261 17 0 00 000002                      PUT     2                       ; save AC2
000013' 200 02 0 00 003171'                     MOVE    2,[CONO]                ; build a CONO
000014' 434 02 0 00 000016                      IOR     2,MBCN                  ; include the device code
000015' 540 02 0 00 000001                      HRR     2,1                     ; include data to write
000016' 256 00 0 00 000002                      XCT     2                       ; do the CONO
000017' 262 17 0 00 000002                      GET     2                       ; restore AC2
000020' 263 17 0 00 000000                      RTN                             ; return
```

```

                                        ;#************************************************************
                                        ;* LDRAR - Load RAR
                                        ;
                                        ; This assumes that 'Diag Test EBUF' bit is not set in CSR, so a DATAO
                                        ; loads the RAR).  The sign bit set in the data specified in the DATAO
                                        ; causes the RAR to be loaded instead of the CRAM.
                                        ;
                                        ; Data to load is in AC1:   Bits 23-34 - CRAM Address
                                        ;                                35    - 0 - Left half of CRAM word
                                        ;                                          1 - Right half of CRAM word
                                        ;
                                        ;#************************************************************

        000021' 332 00 0 00 000010*     LDRAR:  SKIPE   UDEBUG                  ; debug mode?
        000022' 263 17 0 00 000000              RTN                             ; yes - exit
        000023' 261 17 0 00 000001              RPUT    (1,2)                   ; save AC's
        000025' 200 02 0 00 003170'             MOVE    2,[DATAO 0,1]           ; build a DATAO
        000026' 434 02 0 00 000016              IOR     2,MBCN                  ; include the device code
        000027' 242 01 0 00 000026              LSH     1,^D22                  ; position data correctly
        000030' 661 01 0 00 400000              TLO     1,400000                ; set sign bit
        000031' 256 00 0 00 000002              XCT     2                       ; do the DATAO
        000032' 262 17 0 00 000002              RGET    (2,1)                   ; restore AC's
        000034' 263 17 0 00 000000              RTN


                                        ;#************************************************************
                                        ;* LDCRAM - Load CRAM
                                        ;
                                        ; This assumes that 'Diag Test EBUF' bit is not set in CSR, so a
                                        ; DATAO loads the CRAM). The sign bit zero in the data specified
                                        ; in the DATAO causes the CRAM to be loaded instead of the  RAR.
                                        ; The data to load is given right justified in AC1.
                                        ;#************************************************************

        000035' 332 00 0 00 000021*     LDCRAM: SKIPE   UDEBUG                  ; debug mode?
        000036' 263 17 0 00 000000              RTN                             ; yes - exit
        000037' 261 17 0 00 000001              RPUT    (1,2)                   ; save AC's
        000041' 200 02 0 00 003170'             MOVE    2,[DATAO 0,1]           ; build a DATAO
        000042' 434 02 0 00 000016              IOR     2,MBCN                  ; include the device code
        000043' 621 01 0 00 770000              TLZ     1,770000                ; ensure left 6 bits zero
        000044' 256 00 0 00 000002              XCT     2                       ; do the DATAO
        000045' 262 17 0 00 000002              RGET    (2,1)                   ; restore AC's
        000047' 263 17 0 00 000000              RTN
```

```

                                        ;#***************************************************************
                                        ;* Port Register Read Routines
                                        ;
                                        ; Port registers are read by a CONI or DATAI.  The data is
                                        ; returned right justified in AC1 with unused bits zero.
                                        ;
                                        ; Example:  MOVE    AC1,FOO                 ; some argument
                                        ;           GO      RDXXXX                  ; XXXX (Register description)
                                        ;           Return+1                        ; read failed
                                        ;           Return+2                        ; read succeeded
                                        ;
                                        ; Note:  It is not normally possible to tell if the read failed
                                        ;  or succeeded, except with the CSR register.  Reading the CSR
                                        ;  is  set  up  for a +1 and +2 return.  The remaining routines
                                        ;  return +1 always.
                                        ;#***************************************************************

                                        ;#***************************************************************
                                        ;* RDEBUF - Read EBUF
                                        ;
                                        ; This routine assumes that 'Diag Test EBUF' bit is already set
                                        ; in CSR, so a simple DATAI will load the register.
                                        ;#***************************************************************

000050' 332 00 0 00 000035*             RDEBUF: SKIPE   UDEBUG                  ; debug mode?
000051' 263 17 0 00 000000                      RTN                             ; yes - return
000052' 261 17 0 00 000002                      PUT     2                       ; save AC2
000053' 200 02 0 00 003172'                     MOVE    2,[DATAI 0,1]           ; build a DATAI
000054' 434 02 0 00 000016                      IOR     2,MBCN                  ; include the device code
000055' 256 00 0 00 000002                      XCT     2                       ; do the DATAI
000056' 262 17 0 00 000002                      GET     2                       ; restore AC2
000057' 263 17 0 00 000000                      RTN                             ; return


                                        ;#***************************************************************
                                        ;* RDCSR - Read CSR
                                        ;#***************************************************************

000060' 350 00 0 17 000000              RDCSR:  AOS     (P)                     ; set up return+2
000061' 332 00 0 00 000050*                     SKIPE   UDEBUG                  ; debug mode?
000062' 263 17 0 00 000000                      RTN                             ; yes - return
000063' 261 17 0 00 000002                      PUT     2                       ; save AC2
000064' 200 02 0 00 003173'                     MOVE    2,[CONI 0,1]            ; build a CONI from AC1
000065' 434 02 0 00 000016                      IOR     2,MBCN                  ; include the device code
000066' 256 00 0 00 000002                      XCT     2                       ; do the CONI
000067' 255 00 0 00 000000                      JFCL
000070' 255 00 0 00 000000                      JFCL
000071' 336 00 0 00 000001                      SKIPN   1                       ; succeeded?
000072' 370 00 0 17 777777                      SOS     -1(P)                   ; no - set up return+1
000073' 262 17 0 00 000002                      GET     2                       ; yes - error - restore AC2
000074' 263 17 0 00 000000                      RTN                             ;   and return
```

```

                                        ;#**************************************************************
                                        ;* RDLAR - Read LAR
                                        ;
                                        ; This assumes that 'Diag Test EBUF' is not set and 'Diag Sel LAR' is
                                        ; set in the CSR Register, so a DATAI can be used to load the CRAM).
                                        ;
                                        ; The data is returned right justified in AC1:
                                        ;
                                        ;   Bits 23-34 - CRAM Address
                                        ;        35    - 0 - Left half of CRAM word
                                        ;                1 - Right half of CRAM word
                                        ;#**************************************************************

000075' 332 00 0 00 000061*             RDLAR:  SKIPE   UDEBUG                  ; debug mode?
000076' 263 17 0 00 000000                      RTN                             ; yes - return
000077' 261 17 0 00 000002                      PUT     2                       ; save AC2
000100' 200 02 0 00 003172'                     MOVE    2,[DATAI 0,1]           ; build a DATAI from AC1
000101' 434 02 0 00 000016                      IOR     2,MBCN                  ; include the device code
000102' 256 00 0 00 000002                      XCT     2                       ; do the DATAI
000103' 135 01 0 00 003174'                     LDB     1,[POINT 13,1,13]       ; position correctly
000104' 262 17 0 00 000002                      GET     2                       ; restore AC2
000105' 263 17 0 00 000000                      RTN                             ; return


                                        ;#**************************************************************
                                        ;* RDCRAM - Read CRAM
                                        ;
                                        ; This assumes that 'Diag Test EBUF' and 'Diag Sel LAR' bits  are
                                        ; not set in the CSR Register, so a DATAI can be used to load the
                                        ; CRAM).  The data is returned in AC1.
                                        ;#**************************************************************

000106' 332 00 0 00 000075*             RDCRAM: SKIPE   UDEBUG                  ; debug mode?
000107' 263 17 0 00 000000                      RTN                             ; yes - return
000110' 261 17 0 00 000002                      PUT     2                       ; save AC2
000111' 200 02 0 00 003172'                     MOVE    2,[DATAI 0,1]           ; build a DATAI from AC1
000112' 434 02 0 00 000016                      IOR     2,MBCN                  ; include the device code
000113' 256 00 0 00 000002                      XCT     2                       ; do the DATAI
000114' 621 01 0 00 770000                      TLZ     1,770000                ; ensure leftmost 6 bits are zero
000115' 262 17 0 00 000002                      GET     2                       ; restore AC2
000116' 263 17 0 00 000000                      RTN                             ; return
```

```
                                            ;#*****************************************************************
                                            ;* .CONI - Do a simple CONI (no error checking)
                                            ;
                                            ; Argument:  MBCN - Contains RH number to do (left justified properly)
                                            ;                   as 54000,,0, etc.
                                            ;
                                            ; Returns:   AC1 - Contains result
                                            ;            +1 always
                                            ;#*****************************************************************

000117' 332 00 0 00 000106*                 .CONI:  SKIPE   UDEBUG                  ; debug mode?
000120' 263 17 0 00 000000                          RTN                             ; yes - exit
000121' 261 17 0 00 000002                          PUT     2                       ; save AC2
000122' 200 02 0 00 003173'                         MOVE    2,[CONI 0,1]            ; build a CONI from AC1
000123' 434 02 0 00 000016                          IOR     2,MBCN                  ; include the device code
000124' 256 00 0 00 000002                          XCT     2                       ; do the CONI
000125' 262 17 0 00 000002                          GET     2                       ; restore AC2
000126' 263 17 0 00 000000                          RTN                             ; return


                                            ;#*****************************************************************
                                            ;* .CONO - Do a simple CONO
                                            ;
                                            ; Argument:  MBCN - Contains RH number to do (left justified properly)
                                            ;                   as 54000,,0, etc.
                                            ;
                                            ; Returns:   AC1 - Contains data to write to RH or port
                                            ;            +1 always
                                            ;#*****************************************************************

000127' 332 00 0 00 000117*                 .CONO:  SKIPE   UDEBUG                  ; debug mode?
000130' 263 17 0 00 000000                          RTN                             ; yes - exit
000131' 261 17 0 00 000002                          PUT     2                       ; save AC2
000132' 200 02 0 00 003171'                         MOVE    2,[CONO]                ; build a CONO
000133' 434 02 0 00 000016                          IOR     2,MBCN                  ; include the device code
000134' 540 02 0 00 000001                          HRR     2,1                     ; include data to write
000135' 256 00 0 00 000002                          XCT     2                       ; do the CONO
000136' 262 17 0 00 000002                          GET     2                       ; restore AC2
000137' 263 17 0 00 000000                          RTN                             ; return
```

```

                                  ;#**************************************************************
                                  ;* .DATAI - Do a simple DATAI (no error checking)
                                  ;
                                  ; Argument:  MBCN - Contains RH number to do (left justified properly)
                                  ;                   as 54000,,0, etc.
                                  ;
                                  ; Returns:   AC1 - Contains result
                                  ;            +1 always
                                  ;#**************************************************************

000140' 332 00 0 00 000127*       .DATAI: SKIPE   UDEBUG                          ; debug mode?
000141' 263 17 0 00 000000                RTN                                     ; yes - exit
000142' 261 17 0 00 000002                PUT     2                               ; save AC2
000143' 200 02 0 00 003172'               MOVE    2,[DATAI 0,1]                   ; build a DATAI from AC1
000144' 434 02 0 00 000016                IOR     2,MBCN                          ; include the device code
000145' 256 00 0 00 000002                XCT     2                               ; do the DATAI
000146' 262 17 0 00 000002                GET     2                               ; restore AC2
000147' 263 17 0 00 000000                RTN                                     ; return


                                  ;#**************************************************************
                                  ;* .DATAO - Do a simple DATAO
                                  ;
                                  ; Argument:  MBCN - Contains RH number to do (left justified properly)
                                  ;                   as 54000,,0, etc.
                                  ;
                                  ; Returns:   AC1 - Contains data to write to RH or IPA20
                                  ;            +1 always
                                  ;#**************************************************************

000150' 332 00 0 00 000140*       .DATAO: SKIPE   UDEBUG                          ; debug mode?
000151' 263 17 0 00 000000                RTN                                     ; yes - exit
000152' 261 17 0 00 000002                PUT     2                               ; save AC2
000153' 200 02 0 00 003170'               MOVE    2,[DATAO 0,1]                   ; build a DATAO
000154' 434 02 0 00 000016                IOR     2,MBCN                          ; include the device code
000155' 256 00 0 00 000002                XCT     2                               ; do the DATAO
000156' 262 17 0 00 000002                GET     2                               ; restore AC2
000157' 263 17 0 00 000000                RTN                                     ; return
```

```
                                            ;#**********************************************************************
                                            ;* ERESET - Issue an EBUS reset
                                            ;#**********************************************************************

000160' 332 00 0 00 000150*                 ERESET: SKIPE   UDEBUG                  ; debug mode?
000161' 263 17 0 00 000000                          RTN                             ; yes - exit
000162' 336 00 0 00 030037                          SKIPN   USER                    ; user mode?
000163' 7 000 20 0 00 200000                        CONO    APR,200000              ; no - do an EBUS reset
000164' 332 00 0 00 030037                          SKIPE   USER                    ; user mode?
000165' 260 17 0 00 000243'                         GO      IPACLR                  ; yes - do a 'port clear' instead
000166' 263 17 0 00 000000                          RTN
```

```
                                                SUBTTL  Port Specific Utility Routines


                                        ;#*********************************************************************
                                        ;* ISTOP - Stop IPA either by a port clear or by clearing the CSR register
                                        ;#*********************************************************************

000167' 332 00 0 00 000160*     ISTOP:  SKIPE   UDEBUG                          ; debug mode?
000170' 263 17 0 00 000000              RTN                                     ; yes - return
000171' 261 17 0 00 000000              RPUT    (0,1,2)                         ; save AC's
000174' 260 17 0 00 000060'             GO      RDCSR                           ; read CSR register
000175' 474 01 0 00 000000              SETO    1,                              ; error - ensure error bits are set
000176' 200 02 0 00 000001              MOVE    2,1                             ; get into AC2
000177' 603 01 0 00 004000              TLNE    1,(CRAMPE)                      ; CRAM parity error bit set?
000200' 254 00 0 00 000203'             JRST    ISTOP0                          ; yes - must clear it
000201' 606 01 0 00 000010              TRNN    1,MPRUN                         ; IPA running?
000202' 254 00 0 00 000235'             JRST    ISTOPX                          ; no - exit
000203' 200 00 0 00 000001      ISTOP0: MOVE    1                               ; get into AC0
000204' 400 01 0 00 000000              SETZ    1,                              ; CSR data to clear 'MPRUN'
000205' 603 00 0 00 002000              TLNE    (MBERR)                         ; MBUS error bit set?
000206' 254 00 0 00 000231'             JRST    ISTOPC                          ; yes - must do a port clear
000207' 603 00 0 00 004000              TLNE    (CRAMPE)                        ; CRAM parity error bit set?
000210' 201 01 0 00 004000              MOVEI   1,EBUSPE                        ; yes - get 'EBUS PE' bit
000211' 260 17 0 00 000010'             GO      LDCSR                           ; write CSR
000212' 260 17 0 00 000060'             GO      RDCSR                           ; read CSR register
000213' 474 01 0 00 000000              SETO    1,                              ; error - ensure error bits are set
000214' 603 01 0 00 006000              TLNE    1,(CRAMPE!MBERR)                ; error bits set?
000215' 254 00 0 00 000231'             JRST    ISTOPC                          ; yes - must do a port clear
000216' 602 01 0 00 000010              TRNE    1,MPRUN                         ; still running?
000217' 254 00 0 00 000231'             JRST    ISTOPC                          ; yes - must do a port clear
000220' 603 02 0 00 004000              TLNE    2,(CRAMPE)                      ; was CRAM PE set before?
000221' 332 00 0 00 000000*             SKIPE   PMODE                           ; yes - running a test?
000222' 334 00 0 00 000000              SKIPA                                   ; yes - don't print anything
000223' 037 00 0 00 003175'             TMSGCD  <[CRAM PE cleared]>             ; no - let operator know
000224' 607 02 0 00 004000              TLNN    2,(CRAMPE)                      ; was CRAM PE set before?
000225' 332 00 0 00 000221*             SKIPE   PMODE                           ; no - running a test?
000226' 334 00 0 00 000000              SKIPA                                   ; yes - don't print anything
000227' 037 00 0 00 003202'             TMSGCD  <[Port stopped]>                ; no - let operator know
000230' 254 00 0 00 000235'             JRST    ISTOPX                          ; no - exit
000231' 201 01 0 00 400000      ISTOPC: MOVEI   1,PCLEAR                        ; get 'port clear' bit
000232' 260 17 0 00 000010'             GO      LDCSR                           ; write the CSR register
000233' 336 00 0 00 000225*             SKIPN   PMODE                           ; no - running a test?
000234' 037 00 0 00 003206'             TMSGCD  <[Port Clear done]>             ; let operator know
000235' 400 01 0 00 000000      ISTOPX: SETZ    1,                              ; clear AC1
000236' 260 17 0 00 000010'             GO      LDCSR                           ; clear CSR register
000237' 262 17 0 00 000002              RGET    (2,1,0)                         ; restore AC's
000242' 263 17 0 00 000000              RTN                                     ; return
```

```

                                        ;#*******************************************************************
                                        ;* IPACLR - Issue a port clear
                                        ;#*******************************************************************

000243' 261 17 0 00 000001              IPACLR: PUT     1                       ; save AC1
000244' 201 01 0 00 400000                      MOVEI   1,PCLEAR                ; get 'port clear' bit
000245' 260 17 0 00 000010'                     GO      LDCSR                   ; write the CSR register
000246' 262 17 0 00 000001                      GET     1                       ; restore AC1
000247' 263 17 0 00 000000                      RTN                             ; return


                                        ;#*******************************************************************
                                        ;* SETEBU - Check that 'Diag Test EBUF' bit is set in CSR and set if not.
                                        ;
                                        ; Return +1 - Error accessing CSR
                                        ;        +2 - Ok
                                        ;#*******************************************************************

000250' 261 17 0 00 000000              SETEBU: RPUT    (0,1)                   ; save AC's
000252' 260 17 0 00 000060'                     GO      RDCSR                   ; read CSR
000253' 254 00 0 00 000262'                     JRST    SETEBE                  ; error return
000254' 200 00 0 00 000001                      MOVE    1                       ; get copy of AC1 into AC0
000255' 660 01 0 00 200000                      TRO     1,TSTEBF                ; set this bit
000256' 620 01 0 00 040000                      TRZ     1,SELLAR                ; clear this bit
000257' 312 00 0 00 000001                      CAME    1                       ; bits already set up ok?
000260' 260 17 0 00 000010'                     GO      LDCSR                   ; no - set/clear them
000261' 350 00 0 17 777776                      AOS     -2(P)                   ; set up ok return
000262' 262 17 0 00 000001              SETEBE: RGET    (1,0)                   ; restore AC's
000264' 263 17 0 00 000000                      RTN                             ; return


                                        ;#*******************************************************************
                                        ;* SETLAR - Check that 'Diag Sel LAR' bit is set in CSR and set if not.
                                        ;
                                        ; Return +1 - Error accessing CSR
                                        ;        +2 - Ok
                                        ;#*******************************************************************

000265' 261 17 0 00 000000              SETLAR: RPUT    (0,1)                   ; save AC's
000267' 260 17 0 00 000060'                     GO      RDCSR                   ; read CSR
000270' 254 00 0 00 000277'                     JRST    SETLAE                  ; error return
000271' 200 00 0 00 000001                      MOVE    1                       ; get copy of AC1 into AC0
000272' 620 01 0 00 200000                      TRZ     1,TSTEBF                ; clear this bit
000273' 660 01 0 00 040000                      TRO     1,SELLAR                ; set this bit
000274' 312 00 0 00 000001                      CAME    1                       ; bits already set up ok?
000275' 260 17 0 00 000010'                     GO      LDCSR                   ;   is set in CSR Register
000276' 350 00 0 17 777776                      AOS     -2(P)                   ; set up ok return
000277' 262 17 0 00 000001              SETLAE: RGET    (1,0)                   ; restore AC's
000301' 263 17 0 00 000000                      RTN
```

```
 479
 480                                        ;#****************************************************************
 481                                        ;* CLREBU - Check that 'Diag Test EBUF' is not set in CSR and clear if is.
 482                                        ;
 483                                        ; Return +1 - Error accessing CSR
 484                                        ;        +2 - Ok
 485                                        ;#****************************************************************
 486
 487  000302' 261 17 0 00 000000            CLREBU: RPUT    (0,1)                   ; save AC's
 488  000304' 260 17 0 00 000060'                   GO      RDCSR                   ; read CSR
 489  000305' 254 00 0 00 000313'                   JRST    CLREBE                  ; error return
 490  000306' 606 01 0 00 200000                    TRNN    1,TSTEBF                ; 'Diag Test EBUF' bit set?
 491  000307' 254 00 0 00 000312'                   JRST    CLREBX                  ; no - continue
 492  000310' 620 01 0 00 200000                    TRZ     1,TSTEBF                ; make sure 'Diag Test EBUF' bit
 493  000311' 260 17 0 00 000010'                   GO      LDCSR                   ;   is cleared in CSR Register
 494  000312' 350 00 0 17 777776            CLREBX: AOS     -2(P)                   ; set up ok return
 495  000313' 262 17 0 00 000001            CLREBE: RGET    (1,0)                   ; restore AC's
 496  000315' 263 17 0 00 000000                    RTN                             ; return
 497
 498                                        ;#****************************************************************
 499                                        ;* CHKCSR - Check if error bits are set in the CSR register
 500                                        ;
 501                                        ;  Also, stop the port if it is running, and announce the fact to the
 502                                        ;  user.
 503                                        ;#****************************************************************
 504
 505
 506  000316' 261 17 0 00 000001            CHKCSR: PUT     1                       ; save AC1
 507  000317' 260 17 0 00 000060'                   GO      RDCSR                   ; read CSR register
 508  000320' 334 00 0 00 000000                    SKIPA                           ; error return
 509  000321' 603 01 0 00 002000                    TLNE    1,(MBERR)               ; MBUS Error bit set?
 510  000322' 254 00 0 00 000333'                   JRST    CHKCSX                  ; yes - error exit
 511  000323' 603 01 0 00 004000                    TLNE    1,(CRAMPE)              ; CRAM PE bit set?
 512                                                GO      [MOVEI  1,EBUSPE        ; yes - clear it
 513                                                         GO     LDCSR
 514                                                         TMSGCD <[CRAM PE cleared from CSR register]>
 515  000324' 260 17 0 00 003223'                            RTN]
 516  000325' 350 00 0 17 777777                    AOS     -1(P)                   ; no - set up RTN+2
 517
 518                                        ; Also, check to see if the port is running
 519
 520  000326' 606 01 0 00 000010                    TRNN    1,MPRUN                 ; running?
 521  000327' 254 00 0 00 000333'                   JRST    CHKCSX                  ; no - continue
 522  000330' 400 01 0 00 000000                    SETZ    1,                      ; clear AC1
 523  000331' 260 17 0 00 000010'                   GO      LDCSR                   ; stop the port
 524  000332' 037 00 0 00 003227'                   TMSGCD  <[CSR cleared to stop the port]>
 525  000333' 262 17 0 00 000001            CHKCSX: GET     1                       ; restore AC1
 526  000334' 263 17 0 00 000000                    RTN                             ; return
```

```
;#*****************************************************************
;* IPASRT - Start the port at the address given in 'SADDR' and with
;*              initial parameters given in 'SDATA'.
;
; Arguments:    SNEXT - Contains the start address desired.
;
;               SDATA - Contains the data to be written to the CSR
;                 registers to initialize everything correctly.
;
; Note:  This routine is normally used in conjunction with IPASTP.  IPASTP
;        saves the next address in SNEXT and the  contents of CSR in SDATA.
;        Calling IPASRT after IPASTP has been called will leave the port in
;        the same state that it was, unless something is done  to  the port
;        between the calls.
;
; Procedure:    (1) Obtain start address in 'SNEXT' and write it to the
;                   RAR Register.
;
;               (2) If SDATA is zero, no initial data is assumed.  The
;                   CSR is written as follows:
;
;                   Clear 'Single Cycle' - no single stepping apparently
;                     is desired.
;                   Set 'MPROC Run' - Start it up at address in RAR.
;                   Set PIA bits to 4 - Allow interrupts on channel 4.
;                   Clear all remaining bits.
;
;               (3) If SDATA is non-zero, the initial data given in SDATA
;                   is massaged to ensure that at least the following is
;                   true of the data:
;
;                   'Single Cycle' bit is cleared.
;                   'MPROC Run' bit is set.
;
; Return:       +1 - Failed (Error reading/writing CSR Register).
;               +2 - Failed (Port already running).
;               +3 - Failed (Error bits set in CSR)
;               +4 - Ok
;#*****************************************************************
```

```

000335' 261 17 0 00 000000      IPASRT: RPUT    (0,1)                   ; save AC's

                                ; First check if the port is already running

000337' 260 17 0 00 000060'             GO      RDCSR                   ; read CSR
000340' 254 00 0 00 000364'             JRST    IPASR3                  ; error accessing CSR register
000341' 602 01 0 00 000010              TRNE    1,MPRUN                 ; uproc already running?
000342' 254 00 0 00 000363'             JRST    IPASR2                  ; yes - exit +2 (error)

                                ; Check for error bits set

000343' 603 01 0 00 006000              TLNE    1,(CRAMPE!MBERR)        ; error bits set?
000344' 254 00 0 00 000362'             JRST    IPASR1                  ; yes - error exit

                                ; Now handle start address

000345' 622 01 0 00 200000              TRZE    1,TSTEBF                ; 'Diag Test EBUF' bit set?
000346' 260 17 0 00 000010'             GO      LDCSR                   ; yes - clear it
000347' 200 01 0 00 000417'             MOVE    1,SNEXT                 ; get start address
000350' 242 01 0 00 000001              LSH     1,1                     ; position correctly
000351' 260 17 0 00 000021'             GO      LDRAR                   ; load the register

                                ; Set up data to write to CSR

000352' 336 01 0 00 000421'     IPASR0: SKIPN   1,SDATA                 ; any data given?
000353' 201 01 0 00 000014              MOVEI   1,MPRUN!4               ; no - set initial data
000354' 620 01 0 00 020000              TRZ     1,SINCYC                ; ensure SINCYC are cleared
000355' 660 01 0 00 040010              TRO     1,MPRUN!SELLAR          ; ensure MPRUN,SELLAR are set
000356' 332 00 0 00 030037              SKIPE   USER                    ; user mode?
000357' 620 01 0 00 000007              TRZ     1,7                     ; yes - clear PI level bits

                                ; Write the data to CSR

000360' 260 17 0 00 000010'             GO      LDCSR
000361' 350 00 0 17 777776              AOS     -2(P)                   ; ok exit (+4)
000362' 350 00 0 17 777776      IPASR1: AOS     -2(P)                   ; error exit (+3)
000363' 350 00 0 17 777776      IPASR2: AOS     -2(P)                   ; error exit (+2)
000364' 262 17 0 00 000001      IPASR3: RGET    (1,0)                   ; error exit (+1)
000366' 263 17 0 00 000000              RTN                             ; return
```

```
                                     ;#***********************************************************************
                                     ;* IPASTP - Stop the port
                                     ;
                                     ; Arguments:    None
                                     ;
                                     ; Procedure:    Read the CSR Register.
                                     ;               Clear 'MPROC Run' bit.
                                     ;               Write this data to CSR.
                                     ;               Read the CSR Register and save data in SDATA.
                                     ;               Write 'DIAG Sel LAR' bit in CSR.
                                     ;               Read the LAR and save data in SNEXT.
                                     ;
                                     ; Return:       +1 - Failed (Error reading/writing CSR Register).
                                     ;               +2 - Failed (Port not running).
                                     ;               +3 - Failed (Error bits set in CSR)
                                     ;               +4 - Ok, SDATA and SADDR have data at stopping point.
                                     ;#***********************************************************************

000367' 261 17 0 00 000000           IPASTP: RPUT    (0,1)                   ; save AC's

                                     ; First read CSR and verify that 'MPRUN' is set.

000371' 260 17 0 00 000060'                  GO      RDCSR                   ; read CSR
000372' 254 00 0 00 000414'                  JRST    IPAST3                  ; failed - exit +1
000373' 606 01 0 00 000010                   TRNN    1,MPRUN                 ; uproc running?
000374' 254 00 0 00 000413'                  JRST    IPAST2                  ; no - failed - exit +2

                                     ; Check for error bits set

000375' 603 01 0 00 006000                   TLNE    1,(CRAMPE!MBERR)        ; error bits set?
000376' 254 00 0 00 000412'                  JRST    IPAST1                  ; yes - error exit

                                     ; Then clear 'MPRUN'.

000377' 620 01 0 00 000010                   TRZ     1,MPRUN                 ; clear 'MPROC Run' bit
000400' 260 17 0 00 000010'                  GO      LDCSR                   ; write the CSR
000401' 260 17 0 00 000060'                  GO      RDCSR                   ; read CSR
000402' 254 00 0 00 000414'                  JRST    IPAST3                  ; failed - exit
000403' 202 01 0 00 000421'                  MOVEM   1,SDATA                 ; save CSR at halt
```

```
                                        ; Now read the stopping address (in LAR)

000404' 660 01 0 00 040000                      TRO     1,SELLAR                ; set 'Diag Sel LAR/SQR' bit
000405' 260 17 0 00 000010'                     GO      LDCSR                   ; write it
000406' 260 17 0 00 000075'                     GO      RDLAR                   ; read LAR
000407' 242 01 0 00 777777                      LSH     1,-1                    ; get LAR data
000410' 202 01 0 00 000417'                     MOVEM   1,SNEXT                 ; save the data

                                        ; Exit

000411' 350 00 0 17 777776                      AOS     -2(P)                   ; ok exit (+4)
000412' 350 00 0 17 777776              IPAST1: AOS     -2(P)                   ; error exit (+3)
000413' 350 00 0 17 777776              IPAST2: AOS     -2(P)                   ; error exit (+2)
000414' 262 17 0 00 000001              IPAST3: RGET    (1,0)                   ; error exit (+1)
000416' 263 17 0 00 000000                      RTN                             ; return

                                        ; Parameter locations

000417' 000000  000000                  SNEXT:  0                               ; next address
000420' 000000  000000                  SLAST:  0                               ; last address
000421' 000000  000000                  SDATA:  0                               ; contents of CSR at when stopped
000422' 000000  000000                  EADDR:  0                               ; correct stopping address
000423' 000000  000000                  AADDR:  0                               ; actual stopping address
```

```
                                        ;#*****************************************************************
                                        ;* IPASST - Single step the port at the address given in 'SNEXT' and
                                        ;*          with initial parameters given in 'SDATA'.
                                        ;
                                        ; Arguments:     SADDR - Contains the start address desired.
                                        ;
                                        ;                SDATA - Contains the data to be written to the CSR
                                        ;                  registers to initialize everything correctly.
                                        ;
                                        ;                AC1 - 0 - Don't preserve EBUF data
                                        ;                     -1 - Preserve EBUF data
                                        ;
                                        ; Procedure:     (1) Start the port exactly as IPASRT would have, except
                                        ;                    that 'Single Cycle' is set at the same time as
                                        ;                    'MPRUN'.
                                        ;
                                        ;                (2) Successive calls to IPASST do as in (1), except
                                        ;                    that SDATA is already set up and SNEXT is being
                                        ;                    continually copied to SLAST and new contents of
                                        ;                    the LAR are being put into SNEXT.
                                        ;
                                        ; Return:        +1 - Failed (Error reading/writing CSR Register).
                                        ;                +2 - Failed (Port already running).
                                        ;                +3 - Failed (Error bits set in CSR)
                                        ;                +4 - Ok
                                        ;#*****************************************************************

000424' 261 17 0 00 000000              IPASST: RPUT    (0,1)                   ; save AC's
000426' 202 01 0 00 003163'                     MOVEM   1,PRSFLG#               ; save 'Preserve EBUF' flag

                                        ; First check if the port is already running

000427' 260 17 0 00 000060'                     GO      RDCSR                   ; read CSR register
000430' 254 00 0 00 000503'                     JRST    IPASS4                  ; error accessing CSR register
000431' 602 01 0 00 000010                      TRNE    1,MPRUN                 ; uproc already running?
000432' 254 00 0 00 000502'                     JRST    IPASS3                  ; yes - exit +2 (error)

                                        ; Check for error bits set

000433' 603 01 0 00 006000                      TLNE    1,(CRAMPE!MBERR)        ; error bits set?
000434' 254 00 0 00 000501'                     JRST    IPASS2                  ; yes - error exit
```

```

                                  ; Now handle start address

000435' 622 01 0 00 200000                TRZE    1,TSTEBF                ; make sure 'Diag Test EBUF' bit
000436' 260 17 0 00 000010'               GO      LDCSR                   ;    is cleared in CSR Register
000437' 200 01 0 00 000417'               MOVE    1,SNEXT                 ; get start address
000440' 242 01 0 00 000001                LSH     1,1                     ; position correctly
000441' 260 17 0 00 000021'               GO      LDRAR                   ; load the register

                                  ; Set up data to write to CSR

000442' 336 01 0 00 000421'       IPASS0: SKIPN   1,SDATA                 ; any data given?
000443' 201 01 0 00 000014                MOVEI   1,MPRUN!4               ; no - set initial data
000444' 660 01 0 00 060010                TRO     1,SELLAR!MPRUN!SINCYC   ; ensure MPRUN,SELLAR,SINCYC are set
000445' 332 00 0 00 030037                SKIPE   USER                    ; user mode?
000446' 620 01 0 00 000007                TRZ     1,7                     ; yes - clear PI level bits

                                  ; Do the single step

000447' 260 17 0 00 000010'               GO      LDCSR                   ; write the data

                                  ; Read CSR

000450' 260 17 0 00 000060'               GO      RDCSR                   ; read CSR
000451' 255 00 0 00 000000                JFCL                            ; ignore error
000452' 202 01 0 00 000421'               MOVEM   1,SDATA                 ; save it

                                  ; Save EBUF contents

000453' 331 00 0 00 003163'               SKIPL   PRSFLG                  ; preserve EBUF data?
000454' 254 00 0 00 000464'               JRST    IPASS1                  ; no - continue
000455' 660 01 0 00 200000                TRO     1,TSTEBF                ; set 'Diag Test EBUF'
000456' 260 17 0 00 000010'               GO      LDCSR                   ; write it
000457' 260 17 0 00 000050'               GO      RDEBUF                  ; read the EBUF
000460' 202 01 0 00 003165'               MOVEM   1,SAVEBF#               ; save EBUF data
000461' 200 01 0 00 000421'               MOVE    1,SDATA                 ; get CSR data
000462' 620 01 0 00 200000                TRZ     1,TSTEBF                ; set 'Diag Test EBUF'
000463' 260 17 0 00 000010'               GO      LDCSR                   ; write it

                                  ; Read LAR

000464' 200 00 0 00 000417'       IPASS1: MOVE    SNEXT                   ; get original next address
000465' 202 00 0 00 000420'               MOVEM   SLAST                   ; save in last address
000466' 260 17 0 00 000075'               GO      RDLAR                   ; read LAR register
000467' 242 01 0 00 777777                LSH     1,-1                    ; get LAR data
000470' 202 01 0 00 000417'               MOVEM   1,SNEXT                 ; set up new next address
```

```

                                          ; Restore EBUF

000471' 200 01 0 00 000421'                       MOVE    1,SDATA                 ; get CSR data back
000472' 660 01 0 00 200000                        TRO     1,TSTEBF                ; set 'Diag Test EBUF'
000473' 260 17 0 00 000010'                       GO      LDCSR                   ; write it
000474' 200 01 0 00 003165'                       MOVE    1,SAVEBF                ; restore EBUF data
000475' 260 17 0 00 000000'                       GO      LDEBUF                  ; write the EBUF
000476' 200 01 0 00 000421'                       MOVE    1,SDATA                 ; get CSR data back
000477' 260 17 0 00 000010'                       GO      LDCSR                   ; write it

000500' 350 00 0 17 777776                        AOS     -2(P)                   ; ok exit (+4)
000501' 350 00 0 17 777776                IPASS2: AOS     -2(P)                   ; error exit (+3)
000502' 350 00 0 17 777776                IPASS3: AOS     -2(P)                   ; error exit (+2)
000503' 262 17 0 00 000001                IPASS4: RGET    (1,0)                   ; error exit (+1)
000505' 263 17 0 00 000000                        RTN                             ; return

```

```
                                                    SUBTTL  PI System Routines

                                          ;#*********************************************************************
                                          ;* INITPI - Initialize and Turn On PI System
                                          ;
                                          ; In exec mode only.  Turn on PI system with all channels activated.
                                          ; Also, write PI level bits to 0's for all other RH's to ensure that
                                          ; only the port interrupts.
                                          ;#*********************************************************************

000506' 332 00 0 00 030037               INITPI: SKIPE   USER                    ; user mode?
000507' 263 17 0 00 000000                       RTN                             ; yes - exit
000510' 261 17 0 00 000001                       PUT     1                       ; save AC1
000511' 7 004 20 0 00 010000                     CONO    PI,PICLR                ; clear PI system

                                          ; Set PI level bits to zero for each RH slot

000512' 7 540 20 0 00 000000                     CONO    540,0                   ; set PIA level bits to 0
000513' 7 544 20 0 00 000000                     CONO    544,0                   ; set PIA level bits to 0
000514' 7 550 20 0 00 000000                     CONO    550,0                   ; set PIA level bits to 0
000515' 7 554 20 0 00 000000                     CONO    554,0                   ; set PIA level bits to 0
000516' 7 560 20 0 00 000000                     CONO    560,0                   ; set PIA level bits to 0
000517' 7 564 20 0 00 000000                     CONO    564,0                   ; set PIA level bits to 0
000520' 7 570 20 0 00 000000                     CONO    570,0                   ; set PIA level bits to 0
000521' 7 574 20 0 00 000000                     CONO    574,0                   ; set PIA level bits to 0

                                          ; Finally activate all channels and turn on the PI system

000522' 7 004 20 0 00 002177             INITPO: CONO    PI,2177                 ; activate all channels
000523' 7 004 20 0 00 000200                     CONO    PI,PION                 ; turn on PI system
000524' 261 17 0 00 000000                       PUT     0                       ; save AC0
000525' 200 00 0 00 003236'                      MOVE    [INTNUM,,INTNUM+1]      ; build BLT pointer
000526' 251 00 0 00 001327'                      BLT     INTEND                  ; clear interrupt history data
000527' 262 17 0 00 000000                       RGET    (0,1)                   ; restore AC's
000531' 263 17 0 00 000000                       RTN                             ; return

                                          ; Clear PI system after returning from DDT

000532' 332 00 0 00 030037               INITPD: SKIPE   USER                    ; user mode?
000533' 263 17 0 00 000000                       RTN                             ; yes -  exit
000534' 7 004 20 0 00 010000                     CONO    PI,PICLR                ; clear PI system
000535' 254 00 0 00 000522'                      JRST    INITPO                  ; continue
```

```


                                                  ;#*******************************************************************
                                                  ;* .PIOFF - Detach port from PI system
                                                  ;#*******************************************************************

000536' 332 00 0 00 030037        .PIOFF: SKIPE   USER                            ; user mode?
000537' 263 17 0 00 000000                RTN                                     ; yes - return
000540' 261 17 0 00 000001                PUT     1                               ; save AC1
000541' 260 17 0 00 000564'               GO      PISYOF                          ; no - turn off PI system
000542' 260 17 0 00 000060'               GO      RDCSR                           ; read CSR
000543' 255 00 0 00 000000                JFCL                                    ; error return
000544' 620 01 0 00 000007                TRZ     1,7                             ; clear PIA bits in CSR
000545' 260 17 0 00 000010'               GO      LDCSR                           ; write the bit
000546' 262 17 0 00 000001                GET     1                               ; restore AC1
000547' 263 17 0 00 000000                RTN                                     ; exit


                                                  ;#*******************************************************************
                                                  ;* .PION - Attach port to PI system (Channel 4)
                                                  ;#*******************************************************************

000550' 332 00 0 00 030037        .PION:  SKIPE   USER                            ; user mode?
000551' 263 17 0 00 000000                RTN                                     ; yes - exit
000552' 261 17 0 00 000001                PUT     1                               ; save AC1
000553' 260 17 0 00 000564'               GO      PISYOF                          ; turn PI system off
000554' 260 17 0 00 000060'               GO      RDCSR                           ; read CSR
000555' 255 00 0 00 000000                JFCL                                    ; error return
000556' 620 01 0 00 000007                TRZ     1,7                             ; clear PIA bits
000557' 660 01 0 00 000004                TRO     1,4                             ; set PI level 4
000560' 260 17 0 00 000010'               GO      LDCSR                           ; write the bit
000561' 260 17 0 00 000567'               GO      PISYON                          ; turn PI system back on
000562' 262 17 0 00 000001                GET     1                               ; restore AC
000563' 263 17 0 00 000000                RTN                                     ; exit


                                                  ;#*******************************************************************
                                                  ;* PISYOF - Turn PI system off
                                                  ;#*******************************************************************

000564' 336 00 0 00 030037        PISYOF: SKIPN   USER                            ; user mode?
000565' 7 004 20 0 00 000400              CONO    PI,PIOFF                        ; no - turn it off
000566' 263 17 0 00 000000                RTN                                     ; exit


                                                  ;#*******************************************************************
                                                  ;* PISYON - Turn PI system on
                                                  ;#*******************************************************************

000567' 336 00 0 00 030037        PISYON: SKIPN   USER                            ; user mode?
000570' 7 004 20 0 00 002300              CONO    PI,2300                         ; no - turn on PI system
000571' 263 17 0 00 000000                RTN                                     ; and exit
```

```


                                      ;#*********************************************************************
                                      ;* SETVEC - Set up interrupt vectors for an port under test
                                      ;#*********************************************************************

000572' 402 00 0 00 001357'           SETVEC: SETZM   .INWD1                  ; clear special interrupt data
000573' 332 00 0 00 030037                    SKIPE   USER                    ; user mode?
000574' 263 17 0 00 000000                    RTN                             ; yes - exit
000575' 261 17 0 00 000000                    RPUT    (0,1)                   ; save AC's
000577' 200 00 0 00 003237'                   MOVE    [JSR VIINT]             ; get trap instruction (Vectored Int)
000600' 200 01 0 00 002366'                   MOVE    1,ICWA0+5               ; get initial address of log area
000601' 271 01 0 00 000003                    ADDI    1,3                     ; calculate address of 4th word
000602' 270 01 0 00 002275'                   ADD     1,CBASE                 ; add special page number
000603' 202 00 0 01 000000                    MOVEM   (1)                     ; put in 4th word of logout area
000604' 200 01 0 00 002370'                   MOVE    1,ICWA0+7               ; get initial address of log area
000605' 271 01 0 00 000003                    ADDI    1,3                     ; calculate address of 4th word
000606' 270 01 0 00 002275'                   ADD     1,CBASE                 ; add special page number
000607' 202 00 0 01 000000                    MOVEM   (1)                     ; put in 4th word of logout area
000610' 260 17 0 00 000616'                   GO      SET2N                   ; set up 40+2N transfer addresses
000611' 476 00 0 00 001066'                   SETOM   INTNUM                  ; initialize interrupt count
000612' 402 00 0 00 001065'                   SETZM   IDELAY                  ; clear delay time
000613' 262 17 0 00 000001                    RGET    (1,0)                   ; restore AC's
000615' 263 17 0 00 000000                    RTN


                                      ;#*********************************************************************
                                      ;* SET2N - Set up 40 + 2N locations with interrupt dispatch instruction
                                      ;#*********************************************************************

000616' 261 17 0 00 000000            SET2N:  PUT                             ; save AC0
000617' 200 00 0 00 003240'                   MOVE    [JSR    NVIIN1]         ; channel 1
000620' 202 00 0 00 000042                    MOVEM   42
000621' 200 00 0 00 003241'                   MOVE    [JSR    NVIIN2]         ; channel 2
000622' 202 00 0 00 000044                    MOVEM   44
000623' 200 00 0 00 003242'                   MOVE    [JSR    NVIIN3]         ; channel 3
000624' 202 00 0 00 000046                    MOVEM   46
000625' 200 00 0 00 003243'                   MOVE    [JSR    NVIIN4]         ; channel 4
000626' 202 00 0 00 000050                    MOVEM   50
000627' 200 00 0 00 003244'                   MOVE    [JSR    NVIIN5]         ; channel 5
000630' 202 00 0 00 000052                    MOVEM   52
000631' 200 00 0 00 003245'                   MOVE    [JSR    NVIIN6]         ; channel 6
000632' 202 00 0 00 000054                    MOVEM   54
000633' 200 00 0 00 003246'                   MOVE    [JSR    NVIIN7]         ; channel 7
000634' 202 00 0 00 000056                    MOVEM   56
000635' 262 17 0 00 000000                    GET                             ; restore AC0
000636' 263 17 0 00 000000                    RTN
```

```
                                           ;#*********************************************************************
                                           ;* VIINT - Vectored interrupt (Port)
                                           ;
                                           ; This routine (and NVIINx) do the following:
                                           ;
                                           ;   - CONI PI and save it
                                           ;   - Shut off PI system
                                           ;   - Increment interrupt offset (not to exceed 20 interrupts)
                                           ;   - Save interrupt data
                                           ;   - Exit with a JEN after turning PI system back on
                                           ;
                                           ; Interrupt data maintained:
                                           ;
                                           ;   INTPC - PC interrupted from
                                           ;   INTTYP - Type 0,,0 - Vectored                  (If overflow occurred
                                           ;                 0,,n - Non-vectored channel N     the left half is -1)
                                           ;   INTCON - CONI PI data
                                           ;   INTAPR - CONI APR data
                                           ;   INTTIM - Time of interrupt (in usec)
                                           ;   INTUSE - Time of interrupt (addl time in usec)
                                           ;   INTCSR - CSR data on interrrupt
                                           ;   INTNNN - Interrupt #
                                           ;
                                           ; The index into these tables is saved in INTNUM.
                                           ;#*********************************************************************

000637' 000000  000000                     VIINT:  0                                  ; flags and PC get stored here
000640' 7 004 24 0 00 003167'                      CONI    PI,VIIPI#                  ; save CONI PI data
000641' 7 004 20 0 00 000400                       CONO    PI,PIOFF                   ; no - turn off PI system
000642' 261 17 0 00 000000                         RPUT    (0,1,2,3,$SVUUO,$SVUPC)    ; save some stuff

                                           ; Save interrupt data

000650' 400 00 0 00 000000                         SETZ                               ; initialized interrupt type
000651' 350 01 0 00 001066'                        AOS     1,INTNUM                   ; increment interrupt number
000652' 301 01 0 00 000024                         CAIL    1,^D20                     ; exceed maximum?
000653' 505 00 0 00 777777                         HRLI    777777                     ; yes - flag such
000654' 301 01 0 00 000024                         CAIL    1,^D20                     ; exceed maximum?
000655' 201 01 0 00 000023                         MOVEI   1,^D19                     ; yes - limit to 20 entries
000656' 202 00 0 01 001113'                        MOVEM   INTTYP(1)                  ; save interrupt type
000657' 200 00 0 00 001066'                        MOVE    INTNUM                     ; get interrupt number
000660' 202 00 0 01 001303'                        MOVEM   INTNNN(1)                  ; save it
000661' 200 00 0 00 000637'                        MOVE    VIINT                      ; get interrupt PC
000662' 202 00 0 01 001067'                        MOVEM   INTPC(1)                   ; save it
```

```
000663' 200 00 0 00 003167'          MOVE    VIIPI                   ; get CONI PI data
000664' 202 00 0 01 001137'          MOVEM   INTCON(1)               ; save it
000665' 7 000 24 0 00 000000         CONI    APR,0                   ; get CONI APR data
000666' 202 00 0 01 001163'          MOVEM   INTAPR(1)               ; save it
000667' 400 03 0 00 000000           SETZ    3,                      ; set up for getting time
000670' 260 17 0 00 000000*          GO      RUNTME                  ; get current time
000671' 200 00 0 00 000000*          MOVE    DEBTIM                  ; get time in milliseconds
000672' 202 00 0 01 001207'          MOVEM   INTTIM(1)               ; save it
000673' 200 00 0 00 000000*          MOVE    USTIM                   ; get addl time in microseconds
000674' 202 00 0 01 001233'          MOVEM   INTUSE(1)               ; save it

                             ; Exit after turning on PI system (if not refusing them anymore)

000675' 200 00 0 01 001303'          MOVE    INTNNN(1)               ; get interrupt number
000676' 307 00 0 00 020000           CAIG    20000                   ; more than 20000 interrupts?
000677' 260 17 0 00 000550'          GO      .PION                   ; no - turn on PI system
000700' 303 00 0 00 020000           CAILE   20000                   ; more than 20000 interrupts?
000701' 476 00 0 00 001357'          SETOM   .INWD1                  ; yes - indicate such
000702' 262 17 0 00 030114           RGET    ($SVUPC,$SVUUO,3,2,1,0) ; restore stuff
000710' 254 12 1 00 000637'          JEN     @VIINT                  ; release interrupt and return
```

```
                                         ;#*********************************************************************
                                         ;* NVIINx - Non-Vectored interrupt (Port)
                                         ;#*********************************************************************

000711' 000000  000000                   NVIIN1: 0                                  ; for PC+flags
000712' 7 004 24 0 00 003167'                    CONI    PI,VIIPI                   ; save CONI PI data
000713' 7 004 20 0 00 000400                     CONO    PI,PIOFF                   ; turn off PI system
000714' 261 17 0 00 000000                       PUT     0                          ; save AC0
000715' 200 00 0 00 000711'                      MOVE    NVIIN1                     ; get interrupt PC
000716' 202 00 0 00 001064'                      MOVEM   NVJEN                      ; save it for JEN
000717' 201 00 0 00 000001                       MOVEI   1                          ; set up interrupt type
000720' 254 00 0 00 001001'                      JRST    NVIINC                     ; go to common handling

000721' 000000  000000                   NVIIN2: 0                                  ; for PC+flags
000722' 7 004 24 0 00 003167'                    CONI    PI,VIIPI                   ; save CONI PI data
000723' 7 004 20 0 00 000400                     CONO    PI,PIOFF                   ; turn off PI system
000724' 261 17 0 00 000000                       PUT     0                          ; save AC0
000725' 200 00 0 00 000721'                      MOVE    NVIIN2                     ; get interrupt PC
000726' 202 00 0 00 001064'                      MOVEM   NVJEN                      ; save it for JEN
000727' 201 00 0 00 000002                       MOVEI   2                          ; set up interrupt type
000730' 254 00 0 00 001001'                      JRST    NVIINC                     ; go to common handling

000731' 000000  000000                   NVIIN3: 0                                  ; for PC+flags
000732' 7 004 24 0 00 003167'                    CONI    PI,VIIPI                   ; save CONI PI data
000733' 7 004 20 0 00 000400                     CONO    PI,PIOFF                   ; turn off PI system
000734' 261 17 0 00 000000                       PUT     0                          ; save AC0
000735' 200 00 0 00 000731'                      MOVE    NVIIN3                     ; get interrupt PC
000736' 202 00 0 00 001064'                      MOVEM   NVJEN                      ; save it for JEN
000737' 201 00 0 00 000003                       MOVEI   3                          ; set up interrupt type
000740' 254 00 0 00 001001'                      JRST    NVIINC                     ; go to common handling

000741' 000000  000000                   NVIIN4: 0                                  ; for PC+flags
000742' 7 004 24 0 00 003167'                    CONI    PI,VIIPI                   ; save CONI PI data
000743' 7 004 20 0 00 000400                     CONO    PI,PIOFF                   ; turn off PI system
000744' 261 17 0 00 000000                       PUT     0                          ; save AC0
000745' 200 00 0 00 000741'                      MOVE    NVIIN4                     ; get interrupt PC
000746' 202 00 0 00 001064'                      MOVEM   NVJEN                      ; save it for JEN
000747' 201 00 0 00 000004                       MOVEI   4                          ; set up interrupt type
000750' 254 00 0 00 001001'                      JRST    NVIINC                     ; go to common handling

000751' 000000  000000                   NVIIN5: 0                                  ; for PC+flags
000752' 7 004 24 0 00 003167'                    CONI    PI,VIIPI                   ; save CONI PI data
000753' 7 004 20 0 00 000400                     CONO    PI,PIOFF                   ; turn off PI system
000754' 261 17 0 00 000000                       PUT     0                          ; save AC0
000755' 200 00 0 00 000751'                      MOVE    NVIIN5                     ; get interrupt PC
000756' 202 00 0 00 001064'                      MOVEM   NVJEN                      ; save it for JEN
000757' 201 00 0 00 000005                       MOVEI   5                          ; set up interrupt type
000760' 254 00 0 00 001001'                      JRST    NVIINC                     ; go to common handling
```

```

000761' 000000  000000          NVIIN6: 0                                    ; for PC+flags
000762' 7 004 24 0 00 003167'           CONI    PI,VIIPI                     ; save CONI PI data
000763' 7 004 20 0 00 000400            CONO    PI,PIOFF                     ; turn off PI system
000764' 261 17 0 00 000000              PUT     0                            ; save AC0
000765' 200 00 0 00 000761'             MOVE    NVIIN6                       ; get interrupt PC
000766' 202 00 0 00 001064'             MOVEM   NVJEN                        ; save it for JEN
000767' 201 00 0 00 000006              MOVEI   6                            ; set up interrupt type
000770' 254 00 0 00 001001'             JRST    NVIINC                       ; go to common handling

000771' 000000  000000          NVIIN7: 0                                    ; for PC+flags
000772' 7 004 24 0 00 003167'           CONI    PI,VIIPI                     ; save CONI PI data
000773' 7 004 20 0 00 000400            CONO    PI,PIOFF                     ; turn off PI system
000774' 261 17 0 00 000000              PUT     0                            ; save AC0
000775' 200 00 0 00 000771'             MOVE    NVIIN7                       ; get interrupt PC
000776' 202 00 0 00 001064'             MOVEM   NVJEN                        ; save it for JEN
000777' 201 00 0 00 000007              MOVEI   7                            ; set up interrupt type
001000' 254 00 0 00 001001'             JRST    NVIINC                       ; go to common handling

                                ; Save AC's and calculate interrupt number

001001' 261 17 0 00 000001      NVIINC: RPUT    (1,2,3,$SVUUO,$SVUPC)        ; save some stuff
001006' 350 02 0 00 001066'             AOS     2,INTNUM                     ; increment interrupt number
001007' 301 02 0 00 000024              CAIL    2,^D20                       ; exceed maximum?
001010' 505 00 0 00 777777              HRLI    777777                       ; yes - flag such
001011' 301 02 0 00 000024              CAIL    2,^D20                       ; exceed maximum?
001012' 201 02 0 00 000023              MOVEI   2,^D19                       ; yes - limit to 20 entries

                                ; Save interrupt data

001013' 202 00 0 02 001113'             MOVEM   INTTYP(2)                    ; save interrupt type
001014' 200 00 0 00 001066'             MOVE    INTNUM                       ; get interrupt number
001015' 202 00 0 02 001303'             MOVEM   INTNNN(2)                    ; save interrupt number
001016' 200 00 0 00 001064'             MOVE    NVJEN                        ; get interrupt PC
001017' 202 00 0 02 001067'             MOVEM   INTPC(2)                     ; save it
001020' 200 00 0 00 003167'             MOVE    VIIPI                        ; get CONI PI data
001021' 202 00 0 02 001137'             MOVEM   INTCON(2)                    ; save it
001022' 7 000 24 0 00 000000            CONI    APR,0                        ; get CONI APR data
001023' 202 00 0 02 001163'             MOVEM   INTAPR(2)                    ; save it
001024' 260 17 0 00 000060'             GO      RDCSR                        ; read CSR register
001025' 255 00 0 00 000000              JFCL                                 ; ignore error
001026' 202 01 0 02 001257'             MOVEM   1,INTCSR(2)                  ; save data
001027' 400 03 0 00 000000              SETZ    3,                           ; set up for getting time
001030' 260 17 0 00 000670*             GO      RUNTME                       ; get current time
001031' 200 00 0 00 000671*             MOVE    DEBTIM                       ; get time in milliseconds
001032' 202 00 0 02 001207'             MOVEM   INTTIM(2)                    ; save it
001033' 200 00 0 00 000673*             MOVE    USTIM                        ; get addl time in microseconds
001034' 202 00 0 02 001233'             MOVEM   INTUSE(2)                    ; save it
```

```

                                    ; Save special interrupt data if necessary

001035' 400 03 0 00 000000                  SETZ    3,                        ; clear LAR data
001036' 603 01 0 00 004000                  TLNE    1,(CRAMPE)                ; CRAM PI occurred?
                                            GO      [TLZ     1,(CRAMPE)       ; yes - clear it and read
                                                     TRO     1,SELLAR!EBUSPE ;   LAR
                                                     GO      LDCSR
                                                     GO      RDLAR
                                                     MOVE    3,1
001037' 260 17 0 00 003247'                          RTN]
001040' 603 01 0 00 002000                  TLNE    1,(MBERR)                 ; MBUS Error occurred?
                                            GO      [GO      IPACLR           ; yes - clear it and read
                                                     MOVEI   1,SELLAR         ;   LAR
                                                     GO      LDCSR
                                                     GO      RDLAR
                                                     MOVE    3,1
001041' 260 17 0 00 003255'                          RTN]
001042' 607 01 0 00 006000                  TLNN    1,(CRAMPE!MBERR)          ; interrupt bits in CSR set?
001043' 602 01 0 00 007200                  TRNE    1,EBUSPE!FQUERR!MVRERR!RESQAV
                                            GO      [MOVE    1,INTCSR(2)      ; yes - save special interrupt data
                                                     MOVEM   1,.INWD1
                                                     MOVEM   3,.INWD2
001044' 260 17 0 00 003263'                          RTN]

                                    ; Turn back on PI system and dismiss interrupt

001045' 201 00 0 00 001065'                 MOVEI   IDELAY                    ; get delay time
001046' 332 00 0 00 001065'                 SKIPE   IDELAY                    ; specified?
001047' 260 17 0 00 000000*                 GO      ODELAY                    ; yes - delay a bit
001050' 200 00 0 02 001303'                 MOVE    INTNNN(2)                 ; get interrupt number
001051' 307 00 0 00 020000                  CAIG    20000                     ; more than 20000 interrupts?
001052' 260 17 0 00 000550'                 GO      .PION                     ; no - turn on PI system
001053' 303 00 0 00 020000                  CAILE   20000                     ; more than 20000 interrupts?
001054' 476 00 0 00 001357'                 SETOM   .INWD1                    ; yes - indicate such
001055' 262 17 0 00 030114                  RGET    ($SVUPC,$SVUUO,3,2,1,0)   ; restore stuff
001063' 254 12 1 00 001064'                 JEN     @NVJEN                    ; release interrupt and return

001064' 000000  000000              NVJEN:  0

                                    ; Interrupt data storage

001065' 000000  000000              IDELAY: 0
001066' 000000  000000              INTNUM: 0
001067'                             INTPC:  BLOCK   ^D20
001113'                             INTTYP: BLOCK   ^D20
001137'                             INTCON: BLOCK   ^D20
001163'                             INTAPR: BLOCK   ^D20
001207'                             INTTIM: BLOCK   ^D20
001233'                             INTUSE: BLOCK   ^D20
001257'                             INTCSR: BLOCK   ^D20
001303'                             INTNNN: BLOCK   ^D20
001327' 000000  000000              INTEND: 0
```

```
                                      ;#*********************************************************************
                                      ;* .INPNT - Print special interrupt data
                                      ;#*********************************************************************

001330' 336 00 0 00 001357'           .INPNT: SkIPN   .INWD1                     ; any interrupt data saved yet?
001331' 263 17 0 00 000000                    RTN                                ; no - return
001332' 261 17 0 00 000000                    RPUT    (0,1)                      ; save AC's
001334' 200 01 0 00 001357'                   MOVE    1,.INWD1                   ; get CSR data
001335' 316 01 0 00 003267'                   CAMN    1,[-1]                     ; over 20000 interrupts?
                                              JRST    [TMSGCD <[More than 20000 interrupts - PI system shut off]>
001336' 254 00 0 00 003303'                            JRST    .INPNX]
001337' 603 01 0 00 004000                    TLNE    1,(CRAMPE)                 ; CRAM PE?
                                              JRST    [TMSGC  <[CRAM PE at >     ; yes - print such
                                                       MOVE    .INWD2
                                                       LSH     -1
                                                       PNTOCS
                                                       TMSGD   <]>
001340' 254 00 0 00 003310'                            JRST    .INPNX]
001341' 603 01 0 00 002000                    TLNE    1,(MBERR)                  ; MBUS Error?
                                              JRST    [TMSGC  <[MBUS ERR at > ; yes - print such
                                                       MOVE    .INWD2
                                                       LSH     -1
                                                       PNTOCS
                                                       TMSGD   <]>
001342' 254 00 0 00 003322'                            JRST    .INPNX]
001343' 602 01 0 00 004000                    TRNE    1,EBUSPE                   ; EBUS PE?
                                              JRST    [TMSGCD <[EBUS PE]>        ; yes - print such
001344' 254 00 0 00 003333'                            JRST    .INPNX]
001345' 602 01 0 00 002000                    TRNE    1,FQUERR                   ; FQ Error?
                                              JRST    [TMSGCD <[FreeQ Error]> ; yes - print such
001346' 254 00 0 00 003341'                            JRST    .INPNX]
001347' 602 01 0 00 001000                    TRNE    1,MVRERR                   ; DP Error?
                                              JRST    [TMSGCD <[Data Path Error]> ; yes - print such
001350' 254 00 0 00 003350'                            JRST    .INPNX]
001351' 602 01 0 00 000200                    TRNE    1,RESQAV                   ; RESPQ AV?
001352' 037 00 0 00 003352'                   TMSGCD  <[RspQ Avail]>             ; yes - print such

                                      ; Exit

001353' 402 00 0 00 001357'           .INPNX: SETZM   .INWD1                     ; clear initial word
001354' 262 17 0 00 000001                    RGET    (1,0)                      ; restore AC's
001356' 263 17 0 00 000000                    RTN                                ; return

                                      ; Special interrupt data

001357' 000000  000000                .INWD1: 0                                  ; CSR data
001360' 000000  000000                .INWD2: 0                                  ; LAR data

```

```
                                                SUBTTL  Page Fail Handling Routines

                                        ;#*********************************************************************
                                        ;* PFSET - Set up EPT locations 500-502 to handle page fails
                                        ;#*********************************************************************

001361' 332 00 0 00 030037              PFSET:  SKIPE   USER                    ; user mode?
001362' 263 17 0 00 000000                      RTN                             ; yes - return
001363' 261 17 0 00 000000                      PUT     0                       ; save AC0
001364' 402 00 0 00 000500                      SETZM   500                     ; clear error address
001365' 402 00 0 00 000501                      SETZM   501                     ; clear error PC
001366' 200 00 0 00 000502                      MOVE    502                     ; get original dispatch address
001367' 202 00 0 00 001402'                     MOVEM   PFDISP                  ; save it
001370' 201 00 0 00 001422'                     MOVEI   PFCHK                   ; get address of routine to handle
001371' 202 00 0 00 000502                      MOVEM   502                     ;   page fails and set up 502
001372' 402 00 0 00 001410'                     SETZM   PFAIL                   ; init 'page fail occurred' flag
001373' 402 00 0 00 001403'                     SETZM   PFAPR                   ; init APR status at page fail
001374' 402 00 0 00 001404'                     SETZM   PF500                   ; init location 500 at page fail
001375' 402 00 0 00 001405'                     SETZM   PF501                   ; init location 501 at page fail
001376' 402 00 0 00 001406'                     SETZM   PF502                   ; init location 502 at page fail
001377' 402 00 0 00 001407'                     SETZM   PFAC0                   ; init AC0 at page fail
001400' 262 17 0 00 000000                      GET     0                       ; restore AC0
001401' 263 17 0 00 000000                      RTN                             ; return

                                        ; Page fail data area

001402' 000000  000000                  PFDISP: 0                               ; original dispatch address
001403' 000000  000000                  PFAPR:  0                               ; APR status at page fail
001404' 000000  000000                  PF500:  0                               ; location 500 at page fail
001405' 000000  000000                  PF501:  0                               ; location 501 at page fail
001406' 000000  000000                  PF502:  0                               ; location 502 at page fail
001407' 000000  000000                  PFAC0:  0                               ; AC0 at page fail
001410' 000000  000000                  PFAIL:  0                               ; flag that a page fail has occurred

                                        ;#*********************************************************************
                                        ;* PFCLR - Restore EPT locations 500-502 to original values
                                        ;#*********************************************************************

001411' 332 00 0 00 030037              PFCLR:  SKIPE   USER                    ; user mode?
001412' 263 17 0 00 000000                      RTN                             ; yes - return
001413' 261 17 0 00 000000                      PUT     0                       ; save AC0
001414' 402 00 0 00 000500                      SETZM   500                     ; clear error address
001415' 402 00 0 00 000501                      SETZM   501                     ; clear error PC
001416' 200 00 0 00 001402'                     MOVE    PFDISP                  ; get original dispatch address
001417' 202 00 0 00 000502                      MOVEM   502                     ; save it
001420' 262 17 0 00 000000                      GET     0                       ; restore AC0
001421' 263 17 0 00 000000                      RTN                             ; return
```

```
                                        ;#*********************************************************************
                                        ;* PFCHK - Check reason for page fail and dispatch appropriately
                                        ;
                                        ; This routine determines if the page fail is caused by an AR/ARX Parity
                                        ; Error.  If it is, set the PFAIL flag.  If not, dispatch to the original
                                        ; dispatch location.
                                        ;#*********************************************************************

001422' 202 00 0 00 001407'             PFCHK:  MOVEM   PFAC0                   ; save AC0
001423' 7 000 24 0 00 001403'                   CONI    APR,PFAPR               ; get APR status
001424' 200 00 0 00 000500                      MOVE    500                     ; get location 500
001425' 202 00 0 00 001404'                     MOVEM   PF500                   ; save it
001426' 200 00 0 00 000501                      MOVE    501                     ; get location 501
001427' 202 00 0 00 001405'                     MOVEM   PF501                   ; save it
001430' 200 00 0 00 000502                      MOVE    502                     ; get location 502
001431' 202 00 0 00 001406'                     MOVEM   PF502                   ; save it
001432' 135 00 0 00 003356'                     LDB     [POINT 5,500,5]
001433' 302 00 0 00 000036                      CAIE    36
001434' 306 00 0 00 000037                      CAIN    37
001435' 254 00 0 00 001440'                     JRST    PFCHK0

                                        ; Didn't get an AR/ARx Parity Error page fail as expected - exit to
                                        ; subroutine package page fail handler.

001436' 200 00 0 00 001407'                     MOVE    PFAC0                   ; restore AC0
001437' 254 00 1 00 001402'                     JRST    @PFDISP                 ; dispatch

                                        ; Did get an AR/ARx Parity Error page fail - fix the location and exit

001440' 350 00 0 00 001410'             PFCHK0: AOS     PFAIL                   ; increment page fail count
001441' 402 00 1 00 000500                      SETZM   @500                    ; clear the location
001442' 7 000 20 0 00 021400                    CONO    APR,21400               ; clear error bit
001443' 200 00 0 00 001407'                     MOVE    PFAC0                   ; restore AC0
001444' 350 00 0 00 000501                      AOS     501                     ; skip the failing instruction
001445' 254 12 1 00 000501                      JEN     @501                    ; exit
```

```
                                  SUBTTL  Channel Handling Routines

                        ;#*********************************************************************
                        ;* Routines To Interface With A Channel
                        ;
                        ; There are 5 routines available for interfacing with a data channel
                        ; whose characteristic differences become transparent to the user.
                        ;
                        ;         NAME            FUNCTION
                        ;         ----            --------
                        ;         CHINIT          Initialize. re-sync hardware and software
                        ;         GENCCW          Generate a list of CCW's in core
                        ;         GETLOG          Snapshot logout data to a core buffer area
                        ;         LOGPNT          Print channel logout data
                        ;         CCWPNT          Print a CCW list from memory
                        ;
                        ;
                        ; Notes On Initial Control Word Addr:
                        ;
                        ; With each controller is an associated channel #. The channel routines
                        ; default to the following (exec mode) initial control  word  addresses
                        ; for each of the channels.
                        ;
                        ;         TYPE    DEV. CODE       CHAN #          ICW-ADR
                        ;         ----    ---------       ------          -------
                        ;         RH20    540             0               000(ICWA0)
                        ;         RH20    544             1               004(ICWA1)
                        ;         RH20    550             2               010(ICWA2)
                        ;         RH20    554             3               014(ICWA3)
                        ;         RH20    560             4               020(ICWA4)
                        ;         NIPort  564             5               024(ICWA5)
                        ;         RH20    570             6               030(ICWA6)
                        ;         CIPort  574             7               034(ICWA7)
                        ;
                        ; The RH20 zero fill words are located at EPT locations 060 thru 063
                        ; these locations are set to zero by "CHINIT" if running in exec mode.
                        ;#*********************************************************************
```

```
                                        ;#*****************************************************************
                                        ;* CHINIT - Channel Initialization Routine
                                        ;
                                        ; This routine synchronizes software and hardware channel logic.
                                        ;
                                        ; Ex:   MOVE    1,ARG1          ; setup AC1
                                        ;       GO      CHINIT          ; call the routine
                                        ;       Return                  ; ICWA is in AC1
                                        ;
                                        ; AC1:  Specify (AC1)r start addr of channel command buffer
                                        ;       Specify (AC1)l size of the channel command buffer
                                        ;
                                        ; Note:  For figuring out the size of the command  word  buffer  it
                                        ;   should be noted that the GENCCW routine will transfer a maximum
                                        ;   of 512 words per CCW.  Five additional words should be  allowed
                                        ;   for as overhead.
                                        ;
                                        ; 1. Sets up the initial control word with the proper jump
                                        ; 2. Sets up various software params for channel routines
                                        ;#*************************************************************

 001446' 261 17 0 00 000001             CHINIT: RPUT    (1,2)                   ; save AC's
 001450' 554 02 0 00 000001                     HLRZ    2,1                     ; make AC2/final adr,,start adr
 001451' 275 02 0 00 000001                     SUBI    2,1
 001452' 270 02 0 00 000001                     ADD     2,1
 001453' 504 01 0 00 000002                     HRL     1,2
 001454' 261 17 0 00 000001                     PUT     1                       ; save AC1
 001455' 135 01 0 00 003357'                    LDB     1,[POINT 3,MBCN,9]      ; get channel number
 001456' 200 02 0 00 000001                     MOVE    2,1                     ; now to AC2
 001457' 262 17 0 00 000001                     GET     1                       ; restore the AC

 001460' 202 01 0 02 002307'                    MOVEM   1,CADTBL(2)             ; save characteristics of CCW buffer
 001461' 552 01 0 02 002277'                    HRRZM   1,SCLP(2)               ; set up software cmd list pointer
 001462' 200 02 0 02 002361'                    MOVE    2,ICWA0(2)              ; AC2=ICWA
 001463' 270 02 0 00 002275'                    ADD     2,CBASE                 ; causes paging to occur
 001464' 202 02 0 17 777777                     MOVEM   2,-1(P)                 ; becomes AC1 on return
 001465' 336 00 0 00 000167*                    SKIPN   UDEBUG                  ; debug mode?
 001466' 332 00 0 00 030037                     SKIPE   USER                    ; user mode?
 001467' 254 00 0 00 001501'                    JRST    CHINX                   ; yes - exit
```

```
                                      ; Clear 2nd and 3rd words of logout area if in exec mode and place a
                                      ; jump to command list in the EPT. And clear the zero fill locations
                                      ; of the EPT.

001470' 550 01 0 00 000001           CHINO:  HRRZ    1,1                        ; AC1 ==> 0,,buffer address
001471' 434 01 0 00 002276'                  IOR     1,JMPWD                    ; makes a jump CCW
001472' 202 01 0 02 000000                   MOVEM   1,(2)                      ; make ICW a jump to CCW buffer
001473' 402 00 0 02 000001                   SETZM   1(2)                       ; zero the term control word
001474' 402 00 0 02 000002                   SETZM   2(2)                       ; 2nd and 3rd words now cleared
001475' 402 00 0 00 000060                   SETZM   60
001476' 402 00 0 00 000061                   SETZM   61
001477' 402 00 0 00 000062                   SETZM   62
001500' 402 00 0 00 000063                   SETZM   63                         ; clears the zero fill words
001501' 262 17 0 00 000002           CHINX:  RGET    (2,1)                      ; restore AC's
001503' 263 17 0 00 000000                   RTN                                ; return
```

```
;#*********************************************************************
;* GENCCW - Channel Command Word Generator
;
; This routine takes a PDP10 word count and core address and translates
; it to a list of CCW's in a command ring  buffer defined at init time.
; It returns the starting address of the list within the ring, and  the
; # of CCW's for this transfer.
;
; This routine has the following properties:
;   - A maximum of 512 words per CCW is allowed (1 page).
;   - No CCW will transfer accross a page boundry.
;   - The address passed to this routine is the 1st addr in the user
;        buffer that data will be transferred into or out of.
;   - Buffers need not be page aligned in user program.
;
; Call sequence:
;               MOVE    AC3,ARG3                ; a zero fill specifier
;               MOVE    AC2,ARG2                ; pos or neg word count
;               MOVE    AC1,ARG1                ; an addr up to 22 bits
;               GO      GENCCW                  ; build the command list
;               RETURN                          ; returns here
;
; AC1:
;   Contains a 22 bit address specifying where transfer is to start.
;
; AC2:
;   Positive or negative word count for # of words to go to or from
;      the device.  A negative word count on the RH20 will cause the
;      routine to build a reverse CCW list (core addr decrements).
;      appropriate command list will be generated.
;
; AC3:
;   Not meaningfull except for disks.  Other devices should zero AC3.
;      In user mode under TOPS10, this routine  will clear AC3 because
;      zero  fill  is not allowed with TOPS10 DIAG UUO's.  A positive,
;      non-zero, word count will cause the routine to assume that zero
;      fill command words are desired in the list and  the appropriate
;      zero fill words will be generated.  Essentially, zero fill says
;      that the size of the xfer is an even modulo of the value of AC3.
;      Extra words are padded with zero's supplied by the channel on a
;      device  write and extra words are thrown away by the channel on
;      a device read.
;
;      The following  method  is used to calculate the number of words
;      of fill:  FILWDS = AC3 - remainder of [ABS(AC2)/AC3]
;
; RETURN:
;   On return AC1 will contain the following:
;      (AC1)L = # of command words generated by the routine
;      (AC1)R = Points to the first word in the list.
;#*********************************************************************
```

```

                                ;#*********************************************************************
                                ; Functions of the AC's used in GENCCW routine
                                ;  AC   FUNCTION
                                ;  --   --------
                                ;   0   Holds remaining size in words
                                ;   1   CCW's are manufactured in this AC
                                ;   2   Holds the current core adr at any time
                                ;   3   Holds the logical channel number (0-7)
                                ;   4   Holds current software CLP
                                ;   5   Holds last avail buffer addr
                                ;   6   Holds  (Start adr of CCW list,,# of wds in list)
                                ;   7   Holds the read reverse flag
                                ;  10   A scratch AC
                                ;  11   A scratch AC
                                ;  12   A scratch AC
                                ;#*********************************************************************

001504' 332 00 0 00 030037      GENCCW: SKIPE   USER                    ; user mode?
001505' 332 00 0 00 001465*             SKIPE   UDEBUG                  ; special debug mode?
001506' 254 00 0 00 001514'             JRST    GENGO                   ; no - skip this special code
001507' 332 00 0 00 030516              SKIPE   MONTYP                  ; TOPS10 or TOPS20?
001510' 254 00 0 00 001514'             JRST    GENGO                   ; TOPS20 - do normal code
001511' 402 00 0 00 000003              SETZM   3                       ; clear users AC3 (zero fill illegal)
001512' 260 17 0 00 002216'             GO      CHP10                   ; build a special CCW list for TOPS10
001513' 263 17 0 00 000000              RTN                             ; get out - we are done

                                ; Here for either exec mode or user mode under TOPS20

001514' 570 02 0 00 000002      GENGO:  HRRE    2,2                     ; extend sign in case used immed mode
001515' 261 17 0 00 003157'             PUT     FILWDS#                 ; preserve for re-entrant possibility
001516' 261 17 0 00 000010              RPUT    (10,11,12)              ; save AC's
001521' 261 17 0 00 000000              RPUT    (0,1,2,3,4,5,6,7)       ; save AC's

                                ; Set up read reverse flag and take the absolute value of size which
                                ; will be used as the word count ...

001531' 200 00 0 00 000002              MOVE    2                       ; get wc to AC0 where its needed later
001532' 402 00 0 00 000007              SETZM   7                       ; assume forward CCW list
001533' 325 00 0 00 001536'             JUMPGE  0,CZF                   ; jump if list was positive
001534' 476 00 0 00 000007              SETOM   7                       ; set flag for reverse CCW list
001535' 213 00 0 00 000000              MOVNS                           ; makes the word count positive

                                ; Calculate the number of zero fill words required (if any)

001536' 402 00 0 00 003157'     CZF:    SETZM   FILWDS                  ; assume no zero fill required
001537' 322 03 0 00 001556'             JUMPE   3,ACSET                 ; jump if not doing zero fill
001540' 200 11 0 00 000000              MOVE    11,0                    ; get +WC to AC11
001541' 332 00 0 00 000011              SKIPE   11                      ; is the numerator = 0?
001542' 254 00 0 00 001551'             JRST    CZFD                    ; no - proceed with divide operation
```

```

                                        ; Here because the users WC=0 and must now check for and set up for the
                                        ; possibility that he wants to transfer no words to memory  but instead
                                        ; do throw away operations using zero fill only.

001543' 336 00 0 00 000003                      SKIPN   3                       ; were both WC and zero fill spec = 0?
001544' 260 17 0 00 003140'                     GO      FFF                     ; yes - fatal pgm bug
001545' 202 03 0 00 003157'                     MOVEM   3,FILWDS                ; yes - force remainder = divisor
001546' 200 00 0 00 000003                      MOVE    3                       ; use zero fill spec as WC
001547' 402 00 0 00 000002                      SETZM   2                       ; set core addr=0
001550' 254 00 0 00 001557'                     JRST    ACSET1                  ; go off and set up rest of AC's

                                        ; This calculates the number of fill words (if any) when there is also
                                        ; some words being transferred.  The users wc not 0.

001551' 230 11 0 00 000003              CZFD:   IDIV    11,3                    ; divide to get remainder in AC12
001552' 322 12 0 00 001556'                     JUMPE   12,ACSET                ; no fill words if remainder is zero
001553' 200 11 0 00 000003                      MOVE    11,3                    ; get zero fill specifier to AC11
001554' 274 11 0 00 000012                      SUB     11,12                   ; subtract remainder to get fill count
001555' 202 11 0 00 003157'                     MOVEM   11,FILWDS               ; now have number of zero fill words

                                        ; Set up the rest of the AC's for use by this routine and those it calls

001556' 200 02 0 17 777772              ACSET:  MOVE    2,-6(P)                 ; gets the core address
001557' 135 01 0 00 003357'             ACSET1: LDB     1,[POINT 3,MBCN,9]      ; get channel number
001560' 200 03 0 00 000001                      MOVE    3,1                     ; now to AC3
001561' 200 04 0 03 002277'                     MOVE    4,SCLP(3)               ; gets software CLP
001562' 554 05 0 03 002307'                     HLRZ    5,CADTBL(3)             ; gets final buffer addr
001563' 514 06 0 00 000004                      HRLZ    6,4                     ; start of CCW list,,# of words

                                        ; Check direction of transfer and branch to proper CCW builder routine

001564' 260 17 0 00 001610'                     GO      GRINDF                  ; build CCW list

                                        ; Done with the list.  Time to clean up and exit.

001565' 202 04 0 03 002277'             LSTDON: MOVEM   4,SCLP(3)               ; save updated software CLP.
001566' 206 06 0 17 777772                      MOVSM   6,-6(P)                 ; so we can pass back to user
001567' 262 17 0 00 000007                      RGET    (7,6,5,4,3,2,1,0)       ; restore AC's
001577' 262 17 0 00 000012                      RGET    (12,11,10)
001602' 262 17 0 00 003157'                     GET     FILWDS                  ; restore zero fill count
001603' 332 00 0 00 001505*                     SKIPE   UDEBUG                  ; special debug mode?
001604' 263 17 0 00 000000                      RTN                             ; yes - return
001605' 332 00 0 00 030037                      SKIPE   USER                    ; user mode?
001606' 260 17 0 00 002141'                     GO      CHP20                   ; yes - TOPS20 - build DIAG CCW list ..
001607' 263 17 0 00 000000                      RTN                             ; return
```

```

                                  ;#*************************************************************************
                                  ;* GRINDF - Subroutine to build a forward transfer CCW list.
                                  ;
                                  ; This is called from GENCCW and uses the values already in AC's.
                                  ;
                                  ;   0 - Remaining size in words          5 - Last avail buffer addr
                                  ;   1 - Manufactured CCW's               6 - (Start adr of CCW list,,# wds)
                                  ;   2 - Current core adr at any time     7 - Scratch AC
                                  ;   3 - Logical channel number (0-7)    10 - Scratch AC
                                  ;   4 - Current software CLP            11 - Scratch AC
                                  ;#*************************************************************************

001610' 552 02 0 00 000001        GRINDF: HRRZM   2,1                     ; current addr goes into CCW(R)

                                  ; Code to set AC3 to number of words that remain on the page

001611' 135 11 0 00 003360'               LDB     11,[POINT 9,2,35]       ; get the low order 9 bits of address
001612' 201 07 0 00 001000                MOVEI   7,^D512                 ; assume a full page remains
001613' 322 11 0 00 001615'               JUMPE   11,WCCHK                ; page aligned. full page left.
001614' 274 07 0 00 000011                SUB     7,11                    ; not aligned - calc size remaining

                                  ; See if words to go will fit in this CCW and process accordingly

001615' 317 00 0 00 000007        WCCHK:  CAMG    7                       ; will wc fit into current space?
001616' 254 00 0 00 001622'               JRST    WFIT                    ; yes
001617' 137 07 0 00 003361'               DPB     7,[POINT 11,1,13]       ; no - use AC7 for CCW(L)
001620' 274 00 0 00 000007                SUB     7                       ; decrement wc by amount just used
                                  ;********* Next 2 instructions are ommitted for DFPTA tests. ***************
                                  ;         JUMPE   2,.+2                   ; doing a zero fill CCW right now?
                                  ;         ADD     2,7                     ; no - adjust addr by value of AC7
                                  ;********* This causes every CCW to start at same memory location. **********
001621' 254 00 0 00 001633'               JRST    STRE                    ; and off to common code

                                  ; Here because number of wds left will fit in the remaining space

001622' 137 00 0 00 003361'       WFIT:   DPB     [POINT 11,1,13]         ; put wc into CCW(L)
001623' 322 02 0 00 001631'               JUMPE   2,WFIT1                 ; jump if we're doing zero fill
001624' 336 00 0 00 003157'               SKIPN   FILWDS                  ; no zero fill yet.  Any to follow?
001625' 254 00 0 00 001631'               JRST    WFIT1                   ; no - Then this is last CCW
001626' 200 00 0 00 003157'               MOVE    FILWDS                  ; yes - Get count for zero fill words
001627' 402 00 0 00 000002                SETZM   2                       ; zero fill uses an address of 0
001630' 254 00 0 00 001633'               JRST    STRE                    ; off to common code
001631' 661 01 0 00 200000        WFIT1:  TLO     1,(1B1)                 ; sets the last transfer bit in the CCW
001632' 402 00 0 00 000000                SETZM   0                       ; no more words.  WC=0

                                  ; Store CCW from AC1 in buffer, see if done and branch accordingly

001633' 260 17 0 00 001636'       STRE:   GO      CCWST                   ; puts CCW in buffer
001634' 327 00 0 00 001610'               JUMPG   GRINDF                  ; done? no - loop till done
001635' 263 17 0 00 000000                RTN                             ; yes - exit ... AC's are up to date.
```

```
                                        ;#*************************************************************
                                        ;* CCWST - RH20 CCW Store Routine
                                        ;
                                        ; 1. Puts C(AC1) in CCW buffer
                                        ; 2. Does buffer management
                                        ; 3. Counts each CCW entered
                                        ;
                                        ;   The CCW is returned an AC1 as received except for the ''OP-DATA''
                                        ; bit (bit-0) which is set.
                                        ;#*************************************************************

001636' 261 17 0 00 000001              CCWST:  PUT     1                       ; save AC1
001637' 312 04 0 00 000005                      CAME    4,5                     ; are we at last buffer location?
001640' 254 00 0 00 001646'                     JRST    CCWST1                  ; no - go store the word
001641' 200 01 0 00 002276'                     MOVE    1,JMPWD                 ; yes - build a jump CCW
001642' 540 01 0 03 002307'                     HRR     1,CADTBL(3)             ; include start adr of buffer
001643' 202 01 0 04 000000                      MOVEM   1,(4)                   ; put in the last buffer loc
001644' 350 00 0 00 000006                      AOS     6                       ; count the entry
001645' 550 04 0 00 000001                      HRRZ    4,1                     ; reset the CLP to start of buffer
001646' 262 17 0 00 000001              CCWST1: GET     1                       ; recover original CCW
001647' 661 01 0 00 400000                      TLO     1,(1B0)                 ; set op-data bit
001650' 202 01 0 04 000000                      MOVEM   1,(4)                   ; put in the buffer
001651' 271 04 0 00 000001                      ADDI    4,1                     ; update counter/pointer
001652' 271 06 0 00 000001                      ADDI    6,1
001653' 263 17 0 00 000000                      RTN                             ; and exit
```

```
                                        ;#*************************************************************************
                                        ;* GETLOG - Snapshot The Channel Logout Area
                                        ;
                                        ; This routine takes data from the channel logout area and places it in
                                        ; LOGBUF for LOGPNT to print it out.   It zeros the logout area  as  it
                                        ; transfers also.  This routine must be called before calling LOGPNT.
                                        ;#*************************************************************************

001654' 332 00 0 00 001603*             GETLOG: SKIPE   UDEBUG                  ; debug mode?
001655' 263 17 0 00 000000                      RTN                             ; yes - exit
001656' 261 17 0 00 000000                      RPUT    (0,1,2)                 ; save AC's
001661' 135 01 0 00 003357'                     LDB     1,[POINT 3,MBCN,9]      ; get channel number
001662' 200 01 0 01 002361'                     MOVE    1,ICWAO(1)              ; get ICWA
001663' 270 01 0 00 002275'                     ADD     1,CBASE                 ; get EPT address
001664' 332 00 0 00 030037                      SKIPE   USER                    ; user mode?
                                                GO      [MOVE   DEVADR          ; yes - get device address word
                                                         MOVEM  LOGARG+1        ;   and put in argument list
                                                         MOVE   1,[-^D66,,LOGARG]
                                                         XCT    .DIAG           ; get logout data
                                                         GO     DIAGER          ; error
                                                         MOVEI  1,LOGARG+2      ; set up address of logout data
001665' 260 17 0 00 003363'                              RTN]
001666' 201 02 0 00 002317'                     MOVEI   2,LOGBUF                ; get address to save logout data
001667' 504 02 0 00 000001                      HRL     2,1                     ; build BLT word
001670' 251 02 0 00 002322'                     BLT     2,LOGBUF+3              ; move the data
001671' 262 17 0 00 000002                      RGET    (2,1,0)                 ; restore AC's
001674' 263 17 0 00 000000                      RTN                             ; return

                                        ; Argument list for channel status snapshot routine

001675' 000000  000006                  LOGARG: 6                               ; function
001676' 000 00 0 00 000000                      Z                               ; reserved for device addr word
001677'                                         BLOCK   4                       ; reserved for logout data
001703'                                 PHYLST: BLOCK   ^D60                    ; room for physical I/O list
001777'                                         BLOCK   1                       ; a spare location
```

031

```

                                  ;#*********************************************************************
                                  ;* LOGPNT - Channel Logout Printer
                                  ;
                                  ; Routine prints the status of the channel at logout time.  Logout data
                                  ; is printed from a buffered area, therefore, GETLOG should be executed
                                  ; first.
                                  ;#*********************************************************************

002000' 261 17 0 00 000000        LOGPNT: RPUT    (0,1,2,3)               ; save AC's
002004' 037 00 0 00 003372'               TMSGC   <Channel->
002005' 135 01 0 00 003357'               LDB     1,[POINT 3,MBCN,9]      ; get channel number
002006' 200 03 0 00 000001                MOVE    3,1                     ; now to AC3
002007' 200 00 0 00 000003                MOVE    3                       ; ready to print
002010' 037 01 0 00 000000                PNT1                            ; print it
                                          TMSG    <Logout Data
002011' 037 00 0 00 003375'       Loc   Contents>
002012' 260 17 0 00 002022'               GO      ICWPT                   ; print ICW data
002013' 260 17 0 00 002026'               GO      LOSPT                   ; print SW1 data
002014' 260 17 0 00 002043'               GO      TCWPT                   ; prints TCW status
002015' 262 17 0 00 000003                RGET    (3,2,1,0)               ; restore the AC's
002021' 263 17 0 00 000000                RTN                             ; and exit

                                  ; Printer for initial control word & pointer setup routine

002022' 037 00 0 00 003403'       ICWPT:  TMSGC   < 1     ICW: >
002023' 200 00 0 00 002317'               MOVE    LOGBUF                  ; fetch the initial CW
002024' 037 13 0 00 000000                PNTHW                           ; and print it
002025' 263 17 0 00 000000                RTN                             ; and exit

                                  ; RH20 status word-1 printer

002026' 037 00 0 00 003406'       LOSPT:  TMSGC   < 2     SW1: >
002027' 200 01 0 00 002320'               MOVE    1,LOGBUF+1              ; get SW1 word
002030' 404 01 0 00 003411'               AND     1,[777760,,0]           ; mask off unwanted bits
002031' 400 02 0 00 000000                SETZ    2,                      ; initialize text pointer
002032' 603 01 0 00 400000        LOSPT1: TLNE    1,400000                ; bit set?
                                          XCT     [TMSG   < LOGOUT>       ; yes - print text
                                                   TMSG   < M-PAR>
                                                   TMSG   < -ADRPE>
                                                   TMSG   < -WCZRO>
                                                   TMSG   < NXM>
                                                   JFCL
                                                   JFCL
                                                   JFCL
                                                   JFCL
```

032

```
 1665  1666                                                          TMSG    < LSTXER>
 1667                                                        TMSG    < RH-ERR>
 1668                                                        TMSG    < LNG-WC>
 1669                                                        TMSG    < SHT-WC>
 1670  002033' 256 00 0 02 003435'                           TMSG    < OVRRUN>](2)
 1671  002034' 242 01 0 00 000001                  LSH     1,1                              ; shift in next bit
 1672  002035' 350 00 0 00 000002                  AOS     2                                ; point the next text
 1673  002036' 326 01 0 00 002032'                 JUMPN   1,LOSPT1                         ; loop till done
 1674  002037' 037 00 0 00 003453'                 TMSG    <  CLP points to: >
 1675  002040' 135 00 0 00 003457'                 LDB     [POINT 22,LOGBUF+1,35]           ; fetch the actual CLP address
 1676  002041' 037 16 0 00 000003                  PNTOCS                                   ; print in octal
 1677  002042' 263 17 0 00 000000                  RTN                                      ; and exit
 1678
 1679                                      ; Print termination word from the logout area
 1680
 1681  002043' 037 00 0 00 003460'        TCWPT:   TMSGC   < 3    Last Updated CCW: >
 1682  002044' 200 00 0 00 002321'                 MOVE    LOGBUF+2                         ; fetch last CCW
 1683  002045' 037 13 0 00 000000                  PNTHW                                    ; print it
 1684  002046' 037 00 0 00 003466'                 TMSG    <    WC=>
 1685  002047' 135 00 0 00 003470'                 LDB     [POINT  11,LOGBUF+2,13]          ; gets word count
 1686  002050' 037 15 0 00 000000                  PNTDEC                                   ; print it
 1687  002051' 037 00 0 00 003471'                 TMSG    <.  ADR=>
 1688  002052' 135 00 0 00 003473'                 LDB     [POINT  22,LOGBUF+2,35]          ; gets address
 1689  002053' 037 16 0 00 000003                  PNTOCS                                   ; print it in octal
 1690  002054' 263 17 0 00 000000                  RTN
```

```

                                        ;#**********************************************************************
                                        ;* CCWPNT - Channel Command List Printer
                                        ;
                                        ; Address of the list is - CBUF (in exec mode)
                                        ;                          CHARG (under TOPS20)
                                        ;                          CHARG1 (under TOPS10)
                                        ;
                                        ; Calling Sequence:     GO      CCWPNT          ; print
                                        ;                       Return                  ; and return
                                        ;#**********************************************************************

002055' 261 17 0 00 000000              CCWPNT: RPUT    (0,1,2,3,4)             ; save the AC's
002062' 400 03 0 00 000000                      SETZ    3,                      ; clear count of CCW's

                                        ; Get address of list

002063' 201 04 0 00 000000*                     MOVEI   4,CBUF                  ; get address of CCW list
002064' 336 00 0 00 030037                      SKIPN   USER                    ; user mode?
002065' 254 00 0 00 002071'                     JRST    .+4                     ; no - continue
002066' 201 04 0 00 002252'                     MOVEI   4,CHARG1+2              ; yes - get address of TOPS10 list
002067' 332 00 0 00 030516                      SKIPE   MONTYP                  ; TOPS20?
002070' 201 04 0 00 002204'                     MOVEI   4,CHARG+2               ; yes - get address of TOPS20 list

                                        ; Print titles

002071' 037 00 0 00 003474'                     TMSGC   <CCW List:>
002072' 037 00 0 00 003477'                     TMSGC   <  #  Op Code    Word Cnt  Buffer Adr>
002073' 037 00 0 00 003507'                     TMSGC   <  -  -------    --------  ---------->

                                        ; Initialize last CCW printed and number of CCW's skipped - these will
                                        ; be used to determine what to print so as to shorten the CCW list.

002074' 402 00 0 00 003161'                     SETZM   LSTCCW#                 ; clear 'last CCW' printed
002075' 476 00 0 00 003166'                     SETOM   SKPCCW#                 ; init 'number of CCW's skipped'

                                        ; Loop on CCW's

002076' 336 01 0 04 000000              CCWPNO: SKIPN   1,(4)                   ; fetch current CCW - zero?
002077' 254 00 0 00 002132'                     JRST    CCWPNX                  ; yes - done printing
002100' 350 00 0 00 000004                      AOS     4                       ; point to next word
002101' 350 00 0 00 000003                      AOS     3                       ; increment CCW count

                                        ; Check if have to skip a few of these

002102' 312 01 0 00 003161'                     CAME    1,LSTCCW                ; same as 'last CCW printed'?
002103' 254 00 0 00 002110'                     JRST    CCWPN1                  ; no - go print it
002104' 350 00 0 00 003166'                     AOS     SKPCCW                  ; update 'number of CCW's skipped'
002105' 336 00 0 00 003166'                     SKIPN   SKPCCW                  ; 'number of CCW's skipped' zero?
002106' 037 00 0 00 003517'                     TMSGC   < ..>                   ; yes - print this
002107' 254 00 0 00 002076'                     JRST    CCWPNO                  ; loop till done
```

```

                                        ; Print the CCW word

002110' 202 01 0 00 003161'             CCWPN1: MOVEM   1,LSTCCW                ; update 'last CCW printed'
002111' 476 00 0 00 003166'                     SETOM   SKPCCW                  ; init 'number of CCW's skipped'
002112' 037 00 0 00 030242                      PCRL
002113' 001 03 0 00 000003                      PNTOCC  3,3                     ; print CCW number
002114' 135 01 0 00 003521'                     LDB     1,[POINT 3,LSTCCW,2]    ; get op code
                                                XCT     [TMSG   <  Halt      >  ; print it
                                                         TMSG   <  ??1??     >
                                                         TMSG   <  Jump      >
                                                         TMSG   <  ??3??     >
                                                         TMSG   <  Dt-fwd    >
                                                         TMSG   <  Dt-rev    >
                                                         TMSG   <  Dt-fwd-hlt>
002115' 256 00 0 01 003552'                              TMSG   <  Dt-rev-hlt>](1)
002116' 135 00 0 00 003562'                     LDB     [POINT 11,LSTCCW,13]    ; get word count
002117' 006 07 0 00 000000                      PDEC    7,0                     ; print it
002120' 037 00 0 00 003563'                     TMSG    <   >
002121' 135 00 0 00 003564'                     LDB     [POINT 22,LSTCCW,35]    ; get address
002122' 037 10 0 00 000000                      PNTADR                          ; print it

                                        ; Check for end of list

002123' 135 00 0 00 003521'                     LDB     [POINT 3,LSTCCW,2]      ; fetch control part of the CCW
002124' 306 00 0 00 000002                      CAIN    2                       ; is it a jump CCW?
002125' 254 00 0 00 002076'                     JRST    CCWPNO                  ; no - keep looping
002126' 322 00 0 00 002132'                     JUMPE   CCWPNX                  ; (0-2)=000 means halt
002127' 242 00 0 00 777777                      LSH     -1
002130' 302 00 0 00 000003                      CAIE    3                       ; (0-2) 110 or 111 is lst xfer CCW
002131' 254 00 0 00 002076'                     JRST    CCWPNO                  ; no - keep looping

                                        ; Exit

002132' 037 00 0 00 030242              CCWPNX: PCRL                            ; final CRLF
002133' 262 17 0 00 000004                      RGET    (4,3,2,1,0)             ; restore the AC's
002140' 263 17 0 00 000000                      RTN                             ; and exit
```

```

                                      ;#*********************************************************************
                                      ;* CHP20 - Builds a TOPS20 User Mode Channel Program
                                      ;
                                      ; This routine  is  used  to build a channel program using a diagnostic
                                      ; monitor call. This routine is set up to operate with up to 50 channel
                                      ; command words which is the TOPS20 limit.  (1 page per CCW)
                                      ;
                                      ; This routine expects AC1 to be set up in the following way:
                                      ;       AC1(R) = Pointer to first word
                                      ;       AC1(L) = # of words in the list (50 or less)
                                      ;
                                      ; Calling sequence:
                                      ;
                                      ;       MOVE    1,ARG            ; # of words,,pointer to first
                                      ;       GO      CHP20            ; call the routine
                                      ;       RETURN                   ; returns +1 always
                                      ;#*********************************************************************

002141' 261 17 0 00 000004            CHP20:  RPUT    (4,3,2,0,1)             ; save AC's

                                      ; Get the logical chan # to AC3

002146' 135 01 0 00 003357'                   LDB     1,[POINT 3,MBCN,9]      ; get channel number
002147' 200 03 0 00 000001                    MOVE    3,1                     ; now to AC3

                                      ; Range check the word count

002150' 554 00 0 17 000000                    HLRZ    (P)                     ; get count
002151' 303 00 0 00 000062                    CAILE   ^D50                    ; less than or equal to 50
002152' 260 17 0 00 003140'                   GO      FFF                     ; fatal out

                                      ; Word count ok.  Set up pointer & xfer to argument list

002153' 402 00 0 00 000002                    SETZM   2                       ; clear index
002154' 210 01 0 00 000000                    MOVN    1,0                     ; -CNT to AC1
002155' 514 01 0 00 000001                    HRLZ    1,1                     ; -CNT to left half
002156' 540 01 0 17 000000                    HRR     1,(P)                   ; pointer to right half
002157' 200 00 0 01 000000            CHPGM0: MOVE    (1)                     ; fetch a CCW
002160' 202 00 0 02 002204'           CHPGM1: MOVEM   CHARG+2(2)              ; put in argument list
002161' 402 00 0 02 002205'                   SETZM   CHARG+3(2)              ; zero next loc so end of list obvious
002162' 350 00 0 00 000002                    AOS     2                       ; bump store pointer
002163' 253 01 0 00 002157'                   AOBJN   1,CHPGM0                ; loop till done
```

```

                                        ; Set up an execute DIAG monitor call

002164' 200 01 0 00 003156'                    MOVE    1,DEVADR                ; get device address word
002165' 202 01 0 00 002203'                    MOVEM   1,CHARG+1               ; put in argument list
002166' 271 02 0 00 000002                     ADDI    2,2                     ; AC2 now holds # of arguments
002167' 210 02 0 00 000002                     MOVN    2,2                     ; make it negative
002170' 514 01 0 00 000002                     HRLZ    1,2                     ; -count to left hand side AC1
002171' 541 01 0 00 002202'                    HRRI    1,CHARG                 ; pointer to arglist
002172' 256 00 0 00 002405'             C20D:  XCT     .DIAG                   ; DIAG monitor call
002173' 260 17 0 00 002254'                    GO      DIAGER                  ; error
002174' 262 17 0 00 000001                     RGET    (1,0,2,3,4)             ; no error
002201' 263 17 0 00 000000                     RTN

                                        ; Argument list for channel program command

002202' 000000  000004                  CHARG: 4                               ; function code
002203' 000 00 0 00 000000                     Z                               ; reserved for device address word
002204'                                        BLOCK   ^D10                    ; reserved for CCW's

```

```
                                 ;#*******************************************************************
                                 ;* CHP10 - Builds a TOPS10 Channel Command List
                                 ;
                                 ;  This routine is used to build a TOPS10 channel command list using the
                                 ; DIAG UUO monitor call.  Both forward and read reverse command lists are
                                 ; supported.
                                 ;
                                 ;  For a forward list TOPS10 expects 1 word of the form  (-WC,,ADDR-1)
                                 ; where I/O is to take place.  This is done with a DIAG function 4.
                                 ;
                                 ;  For a read reverse list TOPS10 expects 1 word of the form  (-WC,,ADDR)
                                 ; where I/O is to take place.  This is done with a DIAG function 12.
                                 ;
                                 ;  Calling sequence:
                                 ;       MOVE    AC1,ARG1        ; start addr of buffer (first location)
                                 ;       MOVE    AC2,ARG2        ; word count (+means FWD, -means REV)
                                 ;       GO      CHP10           ; call the routine
                                 ;       RTN+1                   ; RTN+1 always.  AC1 contains # of wds in
                                 ;                               ;   CCW list and pointer to the first
                                 ;                               ;   for compatability with CCWPNT
                                 ;#*******************************************************************

002216' 261 17 0 00 000000      CHP10:  RPUT    (0,1,2)                 ; save AC's
002221' 570 02 0 00 000002              HRRE    2,2                     ; extend sign (user maybe used MOVEI)
002222' 321 02 0 00 002231'             JUMPL   2,CWBACK                ; -WC means reverse list...
002223' 201 00 0 00 000004              MOVEI   4                       ; +WC means fwd.  Get function code
002224' 213 02 0 00 000002              MOVNS   2,2                     ; make AC2/-WC,,ADDR-1
002225' 504 02 0 00 000002              HRL     2,2
002226' 540 02 0 00 000001              HRR     2,1
002227' 275 02 0 00 000001              SUBI    2,1                     ; AC2/-WC,,ADDR-1
002230' 254 00 0 00 002234'             JRST    CP10C                   ; to common code
002231' 201 00 0 00 000012      CWBACK: MOVEI   12                      ; -WC means rev.  Get function code
002232' 504 02 0 00 000002              HRL     2,2                     ; make AC2/-WC,,ADDR
002233' 540 02 0 00 000001              HRR     2,1                     ; AC2/-WC,,ADDR
002234' 202 00 0 00 002250'     CP10C:  MOVEM   CHARG1                  ; put function code in table
002235' 202 02 0 00 002252'             MOVEM   2,CHARG1+2              ; put TOPS10 CCW in table
002236' 200 01 0 00 003156'             MOVE    1,DEVADR                ; get device address word
002237' 202 01 0 00 002251'             MOVEM   1,CHARG1+1              ; put in argument table
002240' 200 01 0 00 003565'             MOVE    1,[-3,,CHARG1]          ; get argument list pointer for DIAG
002241' 256 00 0 00 002405'             XCT     .DIAG                   ; execute the monitor call
002242' 260 17 0 00 002254'             GO      DIAGER                  ; error reporter ....
002243' 262 17 0 00 000002              RGET    (2,1,0)                 ; restore AC's
002246' 200 01 0 00 003566'             MOVE    1,[1,,CHARG1+2]         ; for compatability with CCWPNT
002247' 263 17 0 00 000000              RTN                             ; exit

                                ; Argument list for TOPS10 DIAG CCW list builder

002250' 000 00 0 00 000000      CHARG1: Z                               ; function goes here
002251' 000 00 0 00 000000              Z                               ; device addr goes here
002252' 000 00 0 00 000000              Z                               ; single CCW goes here
002253' 000 00 0 00 000000              Z                               ; a spare word
```

```
                                        ;#*********************************************************************
                                        ;* DIAGER - Report Monitor Call Error In User Mode
                                        ;
                                        ; This routine prints an error message stating where and why a
                                        ; diagnostic monitor call has failed.
                                        ;#*********************************************************************

002254' 261 17 0 00 000000              DIAGER: RPUT     (0,1)                  ; save AC's
002256' 202 01 0 00 003164'                     MOVEM    1,SAC1#                ; save the error number
002257' 037 01 0 00 030242                      PCRLF
002260' 037 01 0 00 003567'                     FMSGC    <? Error return from diagnostic monitor call at PC->
002261' 540 00 0 17 777776                      HRR      -2(P)                  ; get pc
002262' 037 17 0 00 000003                      PNTOCF                          ; print it
002263' 037 01 0 00 003602'                     FMSGC    <? Error code=>
002264' 200 00 0 00 003164'                     MOVE     SAC1                   ; get error code
002265' 037 17 0 00 000003                      PNTOCF                          ; print it
002266' 037 01 0 00 003606'                     FMSG     < - >
002267' 260 17 0 00 003027'                     GO       DGEPNT                 ; print DIAG error message
002270' 037 01 0 00 030242                      PCRLF
002271' 262 17 0 00 000000                      RGET     (0,1)                  ; restore AC's
002273' 263 17 0 00 000000                      RTN                             ; return
```

```

                                        ;#*********************************************************************
                                        ;* Some Channel Routine Variables
                                        ;
                                        ; Here are the conditional RH20 parameters necessary for use with the
                                        ; data channel routines.
                                        ;
                                        ; The # of words (maximum) that will be specified in a single
                                        ; channel control word.
                                        ;#*********************************************************************

 002274' 000000  001000                 MAXWD:  ^D512                           ; default to 1 page per xfer word
 002275' 000000  000000                 CBASE:  0                               ; a base for computing ICWA in RH20
                                                                                ;    the 340000 causes paging
 002276' 200000  000000                 JMPWD:  1B1                             ; a KL10 channel jump word
 002277'                                SCLP:   BLOCK   ^D8                     ; 8 software clp's
 002307'                                CADTBL: BLOCK   ^D8                     ; CCW buff params  last adr,,first adr
 002317'                                LOGBUF: BLOCK   ^D32                    ; buffered logout area
 002357' 000000  000004                 CHK:    4                               ; # of logout words per channel
 002360' 777774  002317'                CHCNT:  -4,,LOGBUF                      ; used by GETLOG

                                        ;Defining initial control wd addr for each channel

 002361' 000000  000000                 ICWA0:  0
 002362' 000000  000004                 ICWA1:  4
 002363' 000000  000010                 ICWA2:  10
 002364' 000000  000014                 ICWA3:  14
 002365' 000000  000020                 ICWA4:  20
 002366' 000000  000024                 ICWA5:  24
 002367' 000000  000030                 ICWA6:  30
 002370' 000000  000034                 ICWA7:  34
```

```
                                             ;#*******************************************************************
                                             ;* CHDATA - Obtain channel logout data and inspect results
                                             ;
                                             ; Arguments:  AC1 - Contains expected word count
                                             ;
                                             ; Returns:    +1 - Error occurred
                                             ;             +2 - No error occurred
                                             ;
                                             ;             AC1 - Contains error flags
                                             ;#*******************************************************************

002371' 261 17 0 00 000000                   CHDATA: PUT     0                        ; save AC0
002372' 260 17 0 00 001654'                          GO      GETLOG                   ; get logout data into LOGBUF
002373' 200 01 0 00 003607'                          MOVE    1,[LOGBUF,,CSTWRD]       ; move it from LOGBUF to
002374' 251 01 0 00 000000#                          BLT     1,CSTWRD+3               ;   CSTWRD
002375' 200 01 0 00 000000#                          MOVE    1,CSTWRD+1               ; status in AC5
002376' 621 01 0 00 400017                           TLZ     1,400017                 ; clear 'LOGO' bit / address bits
002377' 620 01 0 00 777777                           TRZ     1,777777                 ; clear address
002400' 430 01 0 00 003610'                          XOR     1,[ADRPE]                ; complement parity error bit
002401' 262 17 0 00 000000                           GET     0                        ; restore AC0
002402' 263 17 0 00 000000                           RTN                              ; return

```

```
                                                SUBTTL  User Mode Support Routines

                                        ;#*************************************************************
                                        ;* DSETUP - Setup Routine For Diagnostic Function
                                        ;
                                        ;  This routine sets up a location called ".DIAG" with the appropriate
                                        ;  JSYS or UUO for a diagnostic monitor call.  It enables privs under
                                        ;  TOPS-20, checks for multiple CPUs under TOPS-10, and exits with the
                                        ;  user IO mode bit set.
                                        ;
                                        ; Calling sequence:
                                        ;       GO DSETUP                       ; call the routine
                                        ;       RETURN                          ; returns + 1 always
                                        ;#*************************************************************

002403' 047 01 0 00 000163      .UUO:   CALLI   1,163                   ; CALLI AC1,163 (DIAG.) for TOPS-10
002404' 104 00 0 00 000530      .JSYS:  JSYS    530                     ; JSYS 530 (DIAG%) for SNARK (TOPS-20)
002405' 260 17 0 00 002437'     .DIAG:  GO      NOSET                   ; a default error reporter

002406' 336 00 0 00 001654*     DSETUP: SKIPN   UDEBUG                  ; special debug mode?
002407' 336 00 0 00 030037              SKIPN   USER                    ; exec mode?
002410' 263 17 0 00 000000              RTN                             ; yes - then exit
002411' 332 00 0 00 030516              SKIPE   MONTYP                  ; Skip if TOPS-10
002412' 254 00 0 00 002417'             JRST    SNKMON                  ; Nope, TOPS-20

                                ; TOPS monitor found (TOPS-10)

002413' 260 17 0 00 002443'     TPSMON: GO      CPUTST                  ; test for dual CPU
002414' 261 17 0 00 002403'             PUT     .UUO                    ; get appropriate UUO
002415' 262 17 0 00 002405'             GET     .DIAG                   ; move into execute location
002416' 254 00 0 00 002422'             JRST    PRVCHK                  ; Check USER IO privs
```

```

                                ; SNARK monitor found (TOPS-20)

002417' 261 17 0 00 002404'     SNKMON: PUT     .JSYS                           ; Get appropriate JSYS
002420' 262 17 0 00 002405'             GET     .DIAG                           ; Set as function to execute
002421' 260 17 0 00 003120'             GO      ENABLE                          ; Enable available privileges

                                ;Here to make sure whoever is running this has USER IO privs.

002422' 260 17 0 00 003075'     PRVCHK: GO      USRION                          ; Turn on user IO mode
002423' 260 17 0 00 002424'             GO      .+1                             ; Push PC onto stack
002424' 262 17 0 00 000001              GET     1                               ; Pop it from stack into AC1
002425' 603 01 0 00 004000              TLNE    1,(1B6)                         ; Is USRIOT bit set?
002426' 263 17 0 00 000000              RTN                                     ; Yes, return

                                ;Can't enter user I/O mode, print message and die.

                                        FMSGC   <? User does not have proper priviledges to run diagnostic!
002427' 037 01 0 00 003611'     ? >
002430' 201 01 0 00 000101              MOVEI   1,.PRIOU                        ; specify terminal as destination
002431' 525 02 0 00 400000              HRLOI   2,.FHSLF                        ; This fork, last error
002432' 402 00 0 00 000003              SETZM   3                               ; must be cleared
002433' 104 00 0 00 000011              ERSTR%                                  ; does an ERSTR JSYS
002434' 255 00 0 00 000000               JFCL                                   ; Ignore error return 1
002435' 255 00 0 00 000000               JFCL                                   ; Ignore error return 2
002436' 260 17 0 00 003140'             GO      FFF                             ; Back to the monitor

                                ; Default error reporter if we try to execute a DIAG function with
                                ; setting up.

002437' 332 00 0 00 002406*     NOSET:  SKIPE   UDEBUG                          ; debug mode?
002440' 254 00 0 00 030064              JRST    CPOPJ1                          ; Yes, give skip return
002441' 037 01 0 00 003626'             FMSGCD  <? Diagnostic attempted a DIAG before initialization>
002442' 260 17 0 00 003140'             GO      FFF
```

```
                                        ;#***************************************************************
                                        ;* CPUTST - Determine Which Of 2 CPU's To Run On
                                        ;
                                        ;    Here if TOPS10 only.  First  find out if we have a dual CPU system.
                                        ; If so, ask user to select the proper one. You must test RH's only that
                                        ; are on the same CPU. This code also sets up to run on the correct one.
                                        ;
                                        ;     Calling sequence:
                                        ;       GO      CPUTST                  ; call the routine
                                        ;       RTN                             ; RTN+1 always
                                        ;#*********************************************************************

002443' 261 17 0 00 000000              CPUTST: RPUT    (0,1,2)                 ; yes - save AC's
002446' 402 00 0 00 003155'                     SETZM   CPUNUM#                 ; Clear remembered CPU
002447' 200 00 0 00 003642'                     MOVE    [17,,11]                ; TABLE=11  ENTRY=17
002450' 047 00 0 00 000041                      GETTAB  0,                      ; table lookup
002451' 260 17 0 00 003140'                     GO      FFF                     ; fatal out
002452' 607 00 0 00 000040                      TLNN    40                      ; dual CPU system?
002453' 254 00 0 00 002474'                     JRST    DPRCX                   ; no - then we can exit
002454' 336 00 0 00 003162'                     SKIPN   MCPUS#                  ; msg flag printed flag set?
                                                FMSGC   <This is a multi-CPU system.  You must choose only 1 to run in.
002455' 037 01 0 00 003643'             Furthermore, all the ports you select must be on that CPU.>
002456' 476 00 0 00 003162'                     SETOM   MCPUS                   ; clear the flag
002457' 037 01 0 00 003674'                     FMSGC   <Select CPU 0 thru 5 <CR> - >
002460' 037 04 0 00 000003                      TTIDEC
002461' 254 00 0 00 002460'                     JRST    .-1                     ; timeout
002462' 331 00 0 00 000000                      SKIPL                           ; less than 0?
002463' 303 00 0 00 000005                      CAILE   5                       ; less than or equal 5?
002464' 254 00 0 00 002457'                     JRST    .-5                     ; invalid - reask
002465' 202 00 0 00 003155'                     MOVEM   CPUNUM                  ; Save CPU to run on
002466' 200 02 0 00 000000                      MOVE    2,0                     ; get # to AC for indexing later
002467' 201 01 0 00 000001                      MOVEI   1,1                     ; set bit mask to assume CPU0
002470' 242 01 0 02 000000                      LSH     1,(2)                   ; shift into position for selected CPU
002471' 505 01 0 00 000014                      HRLI    1,14                    ; set CPU is Function-14
002472' 047 01 0 00 000075                      SETUUO  1,                      ; set to proper CPU
                                                JRST    [FMSGC  <SETCPU UUO failed.  Either wrong CPU selection or>
                                                         FMSGCD <must have wheel privileges.>
002473' 254 00 0 00 003724'                              GO     FFF]            ; fatal out
002474' 262 17 0 00 000002              DPRCX:  RGET    (2,1,0)                 ; restore AC's
002477' 263 17 0 00 000000                      RTN                             ; and exit
```

```
                                              ;#**************************************************************
                                              ;* DEVREL - User Mode Release Routine
                                              ;
                                              ; Does nothing in exec mode.
                                              ;
                                              ; TOPS-20 user mode: For the CI  port, a DIAG JSYS (function  3),
                                              ; is issued to return the port to the monitor.  For the NI  port,
                                              ; an NI  JSYS call  is issued  to return  the device  to  monitor
                                              ; resources.
                                              ;
                                              ; TOPS-10 user mode: For  the both CI and  NI ports, a DIAG  JSYS
                                              ; (function 3), is issued to return the port to the monitor.
                                              ;
                                              ; This routine is different in use  from RH20 disk and tape  uses
                                              ; in that the device (the port) is released only upon exiting the
                                              ; diagnostic.
                                              ;
                                              ; Return:  +1 always
                                              ;#**************************************************************

 002500' 336 00 0 00 002437*          DEVREL: SKIPN   UDEBUG                          ; special debug mode?
 002501' 336 00 0 00 030037                   SKIPN   USER                            ; or exec mode?
 002502' 263 17 0 00 000000                   RTN                                     ; yes - return

                                              ; See if any port is specified

 002503' 400 16 0 00 000000                   SETZ    MBCN,                           ; initialize MBCN
 002504' 332 00 0 00 002673'                  SKIPE   HAVENI                          ; have the NI port?
 002505' 200 16 0 00 000000*                  MOVE    MBCN,PORTNI                     ; yes - set up MBCN
 002506' 332 00 0 00 002672'                  SKIPE   HAVECI                          ; have the CI port?
 002507' 200 16 0 00 000000*                  MOVE    MBCN,PORTCI                     ; yes - set up MBCN
 002510' 322 16 0 00 003226'                  JUMPE   MBCN,[RTN]                      ; any selected? no - return

                                              ; Do the release: first get device address

 002511' 261 17 0 00 000001                   RPUT    (1,2)                           ; save AC's
 002513' 260 17 0 00 [illegible]                   GO      IPACLR                          ; do a port clear

 002514' 200 00 0 00 000016                   MOVE    MBCN                            ; get device code
 002515' 336 00 0 00 030516                   SKIPN   MONTYP                          ; TOPS-20 monitor?
 002516' 242 00 0 00 000003                   LSH     3                               ; no - position all the way left
 002517' 135 01 0 00 003357'                  LDB     1,[POINT 3,MBCN,9]              ; get channel number
 002520' 137 01 0 00 003727'                  DPB     1,[POINT 3,0,23]                ; put into device address
 002521' 202 00 0 00 003156'                  MOVEM   DEVADR#                         ; save it
 002522' 202 00 0 00 002566'                  MOVEM   RELARG+1                        ; place in argument list

                                              ; Dispatch to the required function

 002523' 316 16 0 00 002507*                  CAMN    MBCN,PORTCI                     ; CI port?
 002524' 260 17 0 00 002552'                  GO      RELCI                           ; yes - go release it
 002525' 316 16 0 00 002505*                  CAMN    MBCN,PORTNI                     ; NI port?
 002526' 260 17 0 00 002540'                  GO      RELNI                           ; yes - go release it
```

```

                                    ; Done - clean up

002527' 402 00 0 00 003160'                   SETZM   INDIAG                  ; no longer in DIAG JSYS
002530' 316 16 0 00 002523*                   CAMN    MBCN,PORTCI             ; CI port?
002531' 402 00 0 00 002672'                   SETZM   HAVECI                  ; yes - clear flag
002532' 316 16 0 00 002525*                   CAMN    MBCN,PORTNI             ; NI port?
002533' 402 00 0 00 002673'                   SETZM   HAVENI                  ; yes - clear flag

                                    ; Clean up and exit

002534' 262 17 0 00 000002                    RGET    (2,1)                   ; restore AC's
002536' 254 02 1 00 003730'                   JRSTF   @[1B6+.+1]              ; Turn off USER IOT
002537' 263 17 0 00 000000                    RTN                             ; exit

                                    ; NI port release (TOPS-20 only)

002540' 336 00 0 00 030516          RELNI:    SKIPN   MONTYP                  ; Skip if TOPS-20
002541' 254 00 0 00 002552'                   JRST    RELCI                   ; Nope, release with DIAG on the -10.
002542' 201 01 0 00 000021                    MOVX    1,.EIREL                ; code is "release ownership of nia20"
002543' 542 01 0 00 002572'                   STOR    1,EIFCN,+NIARG          ; put function code into the arg block
002544' 201 01 0 00 000000                    MOVX    1,0                     ; get nia20 # 0
002545' 506 01 0 00 002574'                   STOR    1,EICHN,+NIARG          ; put channel # into the arg block
002546' 200 01 0 00 003731'                   MOVX    1,NIARG                 ; get argument block address
002547' 104 00 0 00 000630          REL1:     NI%                             ; release ownership of the NIA20
                                              ERJMP   [TMSGCD <? NI JSYS failed in attempt to release NIA20>
002550' 320 16 0 00 003744'                            JRST    .+2]
                                    ;         TMSGCD  <[Released NI port]>
002551' 263 17 0 00 000000                    RTN

                                    ; CI port release or NI (TOPS-10 only) release

002552' 476 00 0 00 003160'         RELCI:    SETOM   INDIAG#                 ; useful for debugging crashes
002553' 200 01 0 00 003746'                   MOVE    1,[-2,,RELARG]          ; pointer used
002554' 256 00 0 00 002405'         REL:      XCT     .DIAG                   ;** Overwritten with SKIPA if DMT test
                                              JRST    [MOVEM  1,SAC1          ; error
                                                       FMSGC  <? Release of >
                                                       GO     PRCINI          ; Print "CI" or "NI"
                                                       FMSG   < port failed - >
                                                       GO     DGEPNT          ; print error message
                                                       PCRLF                  ; final CRLF
002555' 254 00 0 00 003757'                            RTN]                   ; exit
                                    ;         TMSGC   <[Released >            ;** Too wordy, removed this
                                    ;         GO      PRCINI                  ; Print "CI" or "NI"
                                    ;         TMSGD   < port]>
002556' 263 17 0 00 000000                    RTN
```

```

                                         ;Here to print "CI" or "NI" depending on MBCN

002557' 316 16 0 00 002530*              PRCINI: CAMN    MBCN,PORTCI              ; CI port?
002560' 037 00 0 00 000103                       PNTCI   "C"                      ; Yes
002561' 316 16 0 00 002532*                      CAMN    MBCN,PORTNI              ; NI port?
002562' 037 00 0 00 000116                       PNTCI   "N"                      ; Yes
002563' 037 00 0 00 000111                       PNTCI   "I"                      ; Print the "I"
002564' 263 17 0 00 000000                       RTN                              ; Return

                                         ; Argument list for release command

002565' 000000  000003                   RELARG: 3                                ; function 3
002566' 000 00 0 00 000000                       Z                                ; reserved for device adr word

                                         ; Argument list for request command

002567' 000000  000002                   REQARG: 2                                ; function 2
002570' 000 00 0 00 000000                       Z                                ; reserved for device address
002571' 001623  416000                           ^D1000*^D60*^D4000               ; 4000 minutes (in milleseconds)

                                         ; NI argument block

002572'                                  NIARG:  BLOCK   100                      ;.EIBMX NI% jsys argument block

                                         ; Miscellaneous data

002672' 000000  000000                   HAVECI: 0                                ; 'have CI port' flag
002673' 000000  000000                   HAVENI: 0                                ; 'have NI port' flag
```

```
                                        ;#*************************************************************
                                        ;* DEVREQ - User Mode Request Routine
                                        ;
                                        ; Does nothing in exec mode.
                                        ;
                                        ; TOPS-20 user  mode:  For  CI  port,  this  routine  executes  a
                                        ; diagnostic monitor call of Function 2 to request the port  from
                                        ; the monitor.  For  NI port,  this routine executes  an NI  JSYS
                                        ; call to request the port from from the monitor.
                                        ;
                                        ; TOPS-10 user  mode:  For  CI  port and  NI  port  this  routine
                                        ; executes a diagnostic monitor call of Function 2 to request the
                                        ; port from the monitor.
                                        ;
                                        ; This routine is different in use  from RH20 disk and tape  uses
                                        ; in that  the  device (the  port)  is requested  only  when  the
                                        ; diagnostic starts up.
                                        ;
                                        ;#*************************************************************

002674' 350 00 0 17 000000              DEVREQ: AOS     (P)                     ; set up proper return
002675' 336 00 0 00 002500*                     SKIPN   UDEBUG                  ; special debug mode?
002676' 336 00 0 00 030037                      SKIPN   USER                    ; or exec mode?
002677' 263 17 0 00 000000                      RTN                             ; yes - return

                                        ; See if no port is specified

002700' 312 16 0 00 002557*                     CAME    MBCN,PORTCI             ; request either the CI port
002701' 316 16 0 00 002561*                     CAMN    MBCN,PORTNI             ;   or the NI port?
002702' 334 00 0 00 000000                      SKIPA                           ; yes - continue
002703' 263 17 0 00 000000                      RTN                             ; no - exit

                                        ; See if any port is specified if already gotten from the system

002704' 316 16 0 00 002700*                     CAMN    MBCN,PORTCI             ; CI port?
                                                JRST    [SKIPE  HAVECI          ; yes - if have the port then
                                                         RTN                    ;   just exit
002705' 254 00 0 00 003766'                              JRST   .+1]
002706' 316 16 0 00 002701*                     CAMN    MBCN,PORTNI             ; NI port?
                                                JRST    [SKIPE  HAVENI          ; yes - if have the port then
                                                         RTN                    ;   just exit
002707' 254 00 0 00 003771'                              JRST   .+1]

                                        ; Release the other port if necessary

002710' 261 17 0 00 000016                      PUT     MBCN                    ; save MBCN
002711' 336 00 0 00 002672'                     SKIPN   HAVECI                  ; have the CI already?
002712' 332 00 0 00 002673'                     SKIPE   HAVENI                  ; have the NI already?
002713' 260 17 0 00 002500'                     GO      DEVREL                  ; yes - release the device
002714' 262 17 0 00 000016                      GET     MBCN                    ; restore MBCN
002715' 260 17 0 00 003075'                     GO      USRION                  ; set USRIOT bit
```

```
                                         ; Get device address

002716' 261 17 0 00 000001                       RPUT     (1,0)                        ; save AC's
002720' 200 00 0 00 000016                       MOVE     MBCN                         ; get device code
002721' 336 00 0 00 030516                       SKIPN    MONTYP                       ; TOPS-20 monitor?
002722' 242 00 0 00 000003                       LSH      3                            ; no - position all the way left
002723' 135 01 0 00 003357'                      LDB      1,[POINT 3,MBCN,9]           ; get channel number
002724' 137 01 0 00 003727'                      DPB      1,[POINT 3,0,23]             ; put into device address
002725' 202 00 0 00 003156'                      MOVEM    DEVADR#                      ; save it
002726' 202 00 0 00 002570'                      MOVEM    REQARG+1                     ; place in argument list

                                         ; Do the required function

002727' 316 16 0 00 002704*                      CAMN     MBCN,PORTCI                  ; CI port?
002730' 254 00 0 00 002744'                      JRST     REQCI                        ; yes - go release it
002731' 316 16 0 00 002706*                      CAMN     MBCN,PORTNI                  ; NI port?
002732' 254 00 0 00 002777'                      JRST     REQNI                        ; yes - go release it
                                                                                       ; Fall through if unknown MBCN

                                         ; Done - clean up and exit

002733' 316 16 0 00 002727*              REQOK:  CAMN     MBCN,PORTCI                  ; requested the CI port?
002734' 476 00 0 00 002672'                      SETOM    HAVECI                       ; yes - set the switch
002735' 316 16 0 00 002731*                      CAMN     MBCN,PORTNI                  ; requested the NI port?
002736' 476 00 0 00 002673'                      SETOM    HAVENI                       ; yes - set the switch
002737' 260 17 0 00 000243'                      GO       IPACLR                       ; do a port clear

002740' 402 00 0 00 003160'              DEVRQX: SETZM    INDIAG                       ; no longer in DIAG JSYS
002741' 262 17 0 00 000000                       RGET     (0,1)                        ; restore AC's
002743' 263 17 0 00 000000                       RTN                                   ; exit
```

```


                                        ; Request the CI port (-10 and -20) or the NI port (-10 only)

002744' 200 01 0 00 003774'            REQCI:  MOVE    1,[-3,,REQARG]                  ; pointer word
002745' 476 00 0 00 003160'                    SETOM   INDIAG                          ; entering DIAG JSYS
002746' 256 00 0 00 002405'            REQ1:   XCT     .DIAG                           ;** Overwritten with SKIPA if DMT test
002747' 334 00 0 00 000000                     SKIPA                                   ; error
002750' 254 00 0 00 002733'                    JRST    REQOK                           ; continue
002751' 370 00 0 17 777776                     SOS     -2(P)                           ; set up failed return
002752' 552 01 0 00 003164'                    HRRZM   1,SAC1                          ; Save error code
002753' 037 01 0 00 003775'                    FMSGC   <% >
002754' 260 17 0 00 002557'                    GO      PRCINI                          ; Print "CI" or "NI"
002755' 037 01 0 00 003776'                    FMSG    < port not selectable - >
002756' 260 17 0 00 003027'                    GO      DGEPNT                          ; print error message
002757' 037 01 0 00 030242                     PCRLF                                   ; Print CRLF
002760' 200 01 0 00 003164'                    MOVE    1,SAC1                          ; Reload the error code
002761' 302 01 0 00 000006                     CAIE    1,6                             ; Port not enabled (TOPS-10)?
002762' 254 00 0 00 002774'                    JRST    REQCI2                          ; No, maybe its the 20
002763' 037 01 0 00 004003'                    FMSG    <% You must use OPR's "CONFIG REMOVE >
002764' 260 17 0 00 002557'                    GO      PRCINI                          ; Print CI or NI
002765' 037 01 0 00 000055                     PNTCIF  "-"                             ; Output dash
002766' 200 02 0 00 003155'                    MOVE    2,CPUNUM                        ; Load CPU number
002767' 037 01 0 02 000060                     PNTCIF  "0"(2)                          ;  and print it
002770' 037 01 0 00 004013'                    FMSG    <" command to test the >
002771' 260 17 0 00 002557'                    GO      PRCINI                          ; Print CI or NI
002772' 037 01 0 00 004020'                    FMSGD   < port>
002773' 254 00 0 00 002740'                    JRST    DEVRQX                          ; and exit

002774' 306 01 0 00 602712             REQCI2: CAIN    1,602712                        ; Port not enabled (TOPS-20)?
                                               FMSGD   <% You must use OPR's "SET PORT CI UNAVAILABLE" command to test the CI port>
002775' 037 01 0 00 004022'
002776' 254 00 0 00 002740'                    JRST    DEVRQX                          ; exit
```

```

                                        ; Request the NI port (TOPS-20 only)

 002777' 336 00 0 00 030516             REQNI:  SKIPN   MONTYP                  ; Skip if TOPS-20 monitor
 003000' 254 00 0 00 002744'                    JRST    REQCI                   ; TOPS-10 monitor, NI same as CI
 003001' 201 01 0 00 000020                     MOVX    1,.EIGET                ; code is "get the NIA20s ownership"
 003002' 542 01 0 00 002572'                    STOR    1,EIFCN,+NIARG          ; put function code into the arg block
 003003' 201 01 0 00 000000                     MOVX    1,0                     ; get nia20 # 0
 003004' 506 01 0 00 002574'                    STOR    1,EICHN,+NIARG          ; put channel # into the arg block
 003005' 200 01 0 00 003731'                    MOVX    1,NIARG                 ; get argument block address
 003006' 104 00 0 00 000630             REQ2:   NI%                             ;** Overwritten with SKIPA if DMT
                                                ERJMP   [TMSGC  <% NI port not selectable - NI JSYS failed to obtain NIA20>
 003007' 320 16 0 00 004056'                             JRST    GETERR]

                                        ; Read the NIA20's state, and proceed only if it's 'OFF' to be sure no one else
                                        ; is using it.  If its not off, first check that no one is using the NI and
                                        ; then use the OPR command SET PORT NI UNAVAILABLE to turn it off.

 003010' 201 01 0 00 000015                     MOVX    1,.EIRCI                ; code is "read channel info"
 003011' 542 01 0 00 002572'                    STOR    1,EIFCN,+NIARG          ; put function code into the arg block
 003012' 201 01 0 00 000000                     MOVX    1,0                     ; get nia20 # 0
 003013' 506 01 0 00 002574'                    STOR    1,EICHN,+NIARG          ; put channel # into the arg block
 003014' 200 01 0 00 003731'                    MOVX    1,NIARG                 ; get argument block address
 003015' 104 00 0 00 000630                     NI%                             ; read the nia20's state
                                                ERJMP   [TMSGC  <% NI% JSYS failed to obtain NIA20 state>
 003016' 320 16 0 00 004071'                             JRST    GETERR]
 003017' 135 01 0 00 004073'                    LOAD    1,EISST,+NIARG          ; get the state
 003020' 306 01 0 00 000011                     CAXN    1,.EISOF                ; is the nia20 off?
 003021' 254 00 0 00 002733'                    JRST    REQOK                   ; Yes
 003022' 037 00 0 00 004074'                    TMSGC   <% NI JSYS indicates port not OFF after we acquired ownership>

                                        ; Here when NIA20 state was not "OFF"

 003023'                                GETERR: TMSGCD  <% You must use OPR's "SET PORT NI UNAVAILABLE" command to test the NI port>
 003023' 037 00 0 00 004111'
 003024' 040 00 0 00 002540'                    CALL    RELNI                   ; release ownership
 003025' 370 00 0 17 777776                     SOS     -2(P)                   ; set up failed return
 003026' 254 00 0 00 002740'                    JRST    DEVRQX                  ; exit
```

```

                                          ;#*********************************************************************
                                          ;* DGEPNT - Decode and print DIAG error
                                          ;#*********************************************************************

003027' 261 17 0 00 000000                DGEPNT: RPUT    (0,1,2,3)               ; save AC's
003033' 336 00 0 00 030516                        SKIPN   MONTYP                  ; TOPS-10 monitor?
003034' 254 00 0 00 003045'                       JRST    DGEPN1                  ; Yes

                                          ;TOPS-20, use the ERSTR% JSYS to output the error.

003035' 201 01 0 00 000101                        MOVEI   1,.PRIOU                ; Output to terminal
003036' 550 02 0 00 003164'                       HRRZ    2,SAC1                  ; Load error code
003037' 505 02 0 00 400000                        HRLI    2,.FHSLF                ; This fork
003040' 400 03 0 00 000000                        SETZ    3,                      ; No limit on string
003041' 104 00 0 00 000011                        ERSTR%                          ; Error code to string
003042' 255 00 0 00 000000                         JFCL                           ; Ignore
003043' 255 00 0 00 000000                         JFCL                           ;  errors
003044' 254 00 0 00 003056'                       JRST    DGEPNX                  ; exit

                                          ;TOPS-10, search table for the error code and publish.

003045' 550 00 0 00 003164'               DGEPN1: HRRZ    0,SAC1                  ; Load the error code
003046' 205 01 0 00 777767                        MOVSI   1,-<T10TLN-1>           ; Load AOBJN pointer

003047' 554 03 0 01 003063'               DGEPN0: HLRZ    3,T10TAB(1)             ; Get entry
003050' 302 00 0 03 000000                        CAIE    0,(3)                   ; Match?
003051' 253 01 0 00 003047'                       AOBJN   1,DGEPN0                ; No, loop
003052' 550 03 0 01 003063'                       HRRZ    3,T10TAB(1)             ; Get addr of TMSG
003053' 037 00 0 03 000000                        PNTMSG  (3)                     ; Output the error
003054' 331 00 0 00 000001                        SKIPL   1                       ; Was the error code found?
003055' 037 13 0 00 000000                        PNTHW                           ; No, print it

003056' 262 17 0 00 000003                DGEPNX: RGET    (3,2,1,0)               ; restore AC's
003062' 263 17 0 00 000000                        RTN                             ; return

                                          ; TOPS-10 Error Messages

003063' 000001  004131'                   T10TAB: 000001,,[ASCIZ /No privileges to execute DIAG. UUO/]
003064' 000002  004140'                           000002,,[ASCIZ /Argument count is too small/]
003065' 000003  004146'                           000003,,[ASCIZ /No such channel/]
003066' 000006  004152'                           000006,,[ASCIZ /NI port not enabled for maintenance use/]
003067' 000013  004162'                           000013,,[ASCIZ /Illegal function/]
003070' 000014  004166'                           000014,,[ASCIZ /Job cannot be virtual/]
003071' 000017  004173'                           000017,,[ASCIZ /Bad argument list/]
003072' 000021  004177'                           000021,,[ASCIZ /No NI port/]
003073' 000023  004202'                           000023,,[ASCIZ /Microcode reload failed/]
003074' 000000  004207'                           000000,,[ASCIZ /Unknown error code: /]
                000012                            T10TLN==.-T10TAB                ;Size of table
```

```
                                        ;#*******************************************************************
                                        ;* USRION/USRIOF - Turn User I/O Privileges On/Off
                                        ;
                                        ;   This routine does nothing in exec mode.  Under TOPS-20  we  do an enable
                                        ; of privileges and then a USER I/O JSYS to get User-IOT set.  Under TOPS10,
                                        ; a TRPSET UUO gets us into User-IOT mode.
                                        ;#*******************************************************************

003075' 336 00 0 00 002675*             USRION: SKIPN   UDEBUG                  ; special debug mode?
003076' 336 00 0 00 030037                      SKIPN   USER                    ; exec mode?
003077' 263 17 0 00 000000                      RTN                             ; yes - return
003100' 261 17 0 00 000000                      RPUT    (0,1)                   ; save AC's
003102' 201 01 0 00 000000                      MOVEI   1,0                     ; Load null TRPSET arg
003103' 336 00 0 00 030516                      SKIPN   MONTYP                  ; TOPS-10?
003104' 047 01 0 00 000025                      TRPSET  1,                      ; Yes, enter user IO mode
003105' 255 00 0 00 000000                       JFCL                           ; Ignore error
003106' 332 00 0 00 030516                      SKIPE   MONTYP                  ; TOPS-20?
003107' 104 00 0 00 000310                      USRIO%                          ; Yes, enter user IO mode
003110' 255 00 0 00 000000                       JFCL                           ; Ignore error
003111' 262 17 0 00 000001                      RGET    (1,0)                   ; restore the AC's
003113' 263 17 0 00 000000                      RTN

                                        ; Code to turn User I/O off

003114' 336 00 0 00 003075*             USRIOF: SKIPN   UDEBUG                  ; special debug mode?
003115' 336 00 0 00 030037                      SKIPN   USER                    ; exec mode?
003116' 263 17 0 00 000000                      RTN                             ; yes - return
003117' 254 02 1 00 004214'                     JRSTF   @[010000,,CPOPJ]        ; modify the flags
```

```

                                                ;#*********************************************************************
                                                ;* ENABLE - Enable Capabilities of This Process
                                                ;
                                                ; First read capabilities and then enable them.  Same as typing ''ENA''
                                                ; from exec level.
                                                ;#*********************************************************************

         003120' 336 00 0 00 003114*    ENABLE: SKIPN   UDEBUG                  ; special debug mode?
         003121' 336 00 0 00 030037             SKIPN   USER                    ; exec mode?
         003122' 263 17 0 00 000000             RTN                             ; yes - exit
         003123' 261 17 0 00 000001             RPUT    (1,2,3)                 ; save AC's
         003126' 201 01 0 00 400000             MOVEI   1,.FHSLF                ; process handle
         003127' 104 00 0 00 000150             RPCAP%                          ; read capabilities
         003130' 320 16 0 00 003131'            ERJMP   .+1                     ; Ignore error
         003131' 200 03 0 00 000002             MOVE    3,2                     ; get capabilities to AC3
         003132' 104 00 0 00 000151             EPCAP%                          ; now enable
         003133' 320 16 0 00 003134'            ERJMP   .+1                     ; ignore errors
         003134' 262 17 0 00 000003             RGET    (3,2,1)                 ; restore AC's
         003137' 263 17 0 00 000000             RTN                             ; exit
```

```
                                                SUBTTL  Miscellaneous Routines

                                        ;#************************************************************
                                        ;* FFF - Routine To Handle Fatals
                                        ;
                                        ; There is no return from this call.  You end up at DDT in exec mode
                                        ; or the monitor if in user mode.
                                        ;
                                        ;   Calling sequence:
                                        ;
                                        ;       GO      FFF                     ; call routine
                                        ;       ..........                      ; pgm has fataled out ....
                                        ;#************************************************************

003140' 260 17 0 00 002500'             FFF:    GO      DEVREL                  ; release device if necessary
003141' 037 01 0 00 004215'                     FMSGC   <Fatal pgm error at PC - >
003142' 550 00 0 17 000000                      HRRZ    (P)                     ; get calling PC
003143' 275 00 0 00 000001                      SUBI    1                       ; gets correct PC
003144' 037 06 0 00 000001                      PNT6F                           ; print it
003145' 037 01 0 00 030242                      PCRLF
003146' 332 00 0 00 030037                      SKIPE   USER                    ; user mode?
003147' 254 00 0 00 003152'                     JRST    .+3                     ; yes
003150' 254 04 0 00 000000*                     HALT    START                   ; halt .. restart
003151' 254 00 0 00 003150*                     JRST    START                   ; to be sure

                                        ; User mode fatal halt

003152' 260 17 0 00 000000*                     GO      .CLOSE                  ; close output files
003153' 260 17 0 00 000000*                     GO      .EXIT                   ; exit the job
003154' 254 00 0 00 003151*                     JRST    START                   ; restart program


                                        ;#*****************************************************************
                                        ; End of DFPTAI.MAC Module
                                        ;#*****************************************************************

                                                END

NO ERRORS DETECTED

PROGRAM BREAK IS 004223
CPU TIME USED 00:10.026

113P CORE USED
```

```
055
DFPTAI   MACRO %53B(1242) 10:15  1-Jul-85 Page S-1
DFPTAI   MAC      17-May-85 22:17           SYMBOL TABLE                                                    SEQ 2068

AADDR            000423' int    DSETUP           002406' ent    IPASS0           000442'        PF500            001404' int
ACSET            001556'        EADDR            000422' int    IPASS1           000464'        PF501            001405' int
ACSET1           001557'        EBUSPE           004000  spd    IPASS2           000501'        PF502            001406' int
ADRPE    100000  000000  spd    EICHN    777777  000000  spd    IPASS3           000502'        PFACO            001407'
APR              000000  spd    EIFCN            777777  spd    IPASS4           000503'        PFAIL            001410' int
C20D             002172'        EISST            777000  spd    IPASST           000424' ent    PFAPR            001403' int
CADTBL           002307'        ENABLE           003120'        IPAST1           000412'        PFCHK            001422'
CALL     200000  000000  spd    EPCAP%   104000  000151  int    IPAST2           000413'        PFCHKO           001440'
CDASE            002275' int    ERESET           000160' ent    IPAST3           000414'        PFCLR            001411' ent
CBUF             002063' ext    ERJMP    320700  000000  int    IPASTP           000367' ent    PFDISP           001402'
CCWPN0           002076'        ERSTR%   104000  000011  int    ISTOP            000167' ent    PFSET            001361' ent
CCWPN1           002110'        FFF              003140' ent    ISTOP0           000203'        PHYLST           001703'
CCWPNT           002055' ent    FILWDS           003157'        ISTOPC           000231'        PI               000004  spd
CCWPNX           002132'        FQUERR           002000  spd    ISTOPX           000235'        PICLR            010000  spd
CCWST            001636'        GENCCW           001504' ent    JEN      254500  000000         PIOFF            000400  spd
CCWST1           001646'        GENGO            001514'        JMPWD            002276'        PION             000200  spd
CHARG            002202'        GET      262740  000000         JRSTF    254100  000000         PISYOF           000564' ent
CHARG1           002250'        GETERR           003023'        JSYS     104000  000000         PISYON           000567' ent
CHCNT            002360'        GETLOG           001654' ent    LDCRAM           000035' ent    PMODE            000233' ext
CHDATA           002371' ent    GETTAB   047000  000041         LDCSR            000010' ent    PNT1     037040  000000
CHINO            001470'        GO       260740  000000         LDEBUF           000000' ent    PNT6F    037300  000001
CHINIT           001446' ent    GRINDF           001610'        LDRAR            000021' ent    PNTADR   037400  000000
CHINX            001501'        HALT     254200  000000         LOGARG           001675'        PNTCI    037000  000000
CHK              002357'        HAVECI           002672' int    LOGBUF           002317' int    PNTCIF   037040  000000
CHKCSR           000316' ent    HAVENI           002673' int    LOGPNT           002000' ent    PNTDEC   037640  000000
CHKCSX           000333'        ICWA0            002361'        LOSPT            002026'        PNTHW    037540  000000
CHP10            002216'        ICWA1            002362'        LOSPT1           002032'        PNTMSF   037040  000000
CHP20            002141'        ICWA2            002363'        LSTCCW           003161'        PNTMSG   037000  000000
CHPGM0           002157'        ICWA3            002364'        LSTDON           001565'        PNTOCC   001000  000000
CHPGM1           002160'        ICWA4            002365'        MAXWD            002274'        PNTOCF   037740  000003
CISEL            000000  ext    ICWA5            002366'        MBCN             000016         PNTOCS   037700  000003
CLREBE           000313'        ICWA6            002367'        MBERR    002000  000000  spd    PORTCI           002733' ext
CLREBU           000302' ent    ICWA7            002370'        MCPUS            003162'        PORTNI           002735' ext
CLREBX           000312'        ICWPT            002022'        MONTYP           030516         PRCINI           002557'
CP10C            002234'        IDELAY           001065' int    MPRUN            000010  spd    PRSFLG           003163'
CPOPJ            030065         INDIAG           003160'        MVRERR           001000  spd    PRVCHK           002422'
CPOPJ1           030064         INITPO           000522'        NI%      104000  000630  int    PUT      261740  000000
CPUNUM           003155'        INITPD           000532' ent    NIARG            002572'        RDCRAM           000106' ent
CPUTST           002443'        INITPI           000506' ent    NISEL            000000  ext    RDCSR            000060' ent
CRAMPE   004000  000000  spd    INTAPR           001163' int    NOSET            002437'        RDEBUF           000050' ent
CSTWRD           003607' ext    INTCON           001137' int    NVIIN1           000711'        RDLAR            000075' ent
CWBACK           002231'        INTCSR           001257' int    NVIIN2           000721'        REL              002554' ent
CZF              001536'        INTEND           001327' int    NVIIN3           000731'        REL1             002547'
CZFD             001551'        INTNNN           001303'        NVIIN4           000741'        RELARG           002565'
DEBTIM           001031' ext    INTNUM           001066' int    NVIIN5           000751'        RELCI            002552'
DEVADR           003156'        INTPC            001067' int    NVIIN6           000761'        RELNI            002540'
DEVREL           002500' ent    INTTIM           001207' int    NVIIN7           000771'        REQ1             002746' ent
DEVREQ           002674' ent    INTTYP           001113' int    NVIINC           001001'        REQ2             003006' ent
DEVRQX           002740'        INTUSE           001233' int    NVJEN            001064'        REQARG           002567'
DGEPN0           003047'        IPACLR           000243' ent    ODELAY           001047' ext    REQCI            002744'
DGEPN1           003045'        IPASR0           000352'        P                000017         REQCI2           002774'
DGEPNT           003027'        IPASR1           000362'        PCLEAR           400000  spd    REQNI            002777'
DGEPNX           003056'        IPASR2           000363'        PCRL     037000  030242         REQOK            002733'
DIAGER           002254'        IPASR3           000364'        PCRLF    037040  030242         RESQAV           000200  spd
DPRCX            002474'        IPASRT           000335' ent    PDEC     006000  000000         RPCAP%   104000  000150  int
```

```
056
DFPTAI   MACRO %53B(1242) 10:15  1-Jul-85 Page S-2
DFPTAI   MAC      17-May-85 22:17           SYMBOL TABLE                                                    SEQ 2069
DFPTAI  MACRO %53B(1242) 10:15  1-Jul-85 Page S-2
DFPTAI  MAC        17-May-85 22:17           SYMBOL TABLE

RTN      263740  000000           .DATAO          000150' ent
RUNTME           001030' ext      .DIAG           002405'
SAC1             003164'          .EICHN          000002  sin
SAVEBF           003165'          .EIFCN          000000  sin
SCLP             002277'          .EIGET          000020  sin
SDATA            000421' int      .EIRCI          000015  sin
SELLAR           040000  spd      .EIREL          000021  sin
SET2N            000615'          .EISOF          000011  sin
SETEBE           000262'          .EISTA          000004  sin
SETEBU           000250' ent      .EXIT           003153' ext
SETLAE           000277'          .FHSLF          400000  sin
SETLAR           000265' ent      .INPNT          001330' ent
SETUUO   047000  000075           .INPNX          001353'
SETVEC           000572' ent      .INWD1          001357' int
SINCYC           020000  spd      .INWD2          001360' int
SKPCCW           003166'          .JSYS           002404'
SLAST            000420' int      .PIOFF          000536' ent
SNEXT            000417' int      .PION           000550' ent
SNKMON           002417'          .PRIOU          000101  sin
START            003154' ext      .UUO            002403'
STRE             001633'
SWITT            000000  ext
T10TAB           003063'
T10TLN           000012  spd
TCWPT            002043'
TPSMON           002413'
TRPSET   047000  000025
TSTEBF           200000  spd
TTIDEC   037200  000003
UDEBUG           003120' ext
USER             030037
USRIO%   104000  000310  int
USRIOF           003114' ent
USRION           003075' ent
USTIM            001033' ext
VIINT            000637'
VIIPI            003167'
WCCHK            001615'
WFIT             001622'
WFIT1            001631'
ZI               000000'
$EICHN           000002  spd
$EIFCN           000000  spd
$EISST           000004  spd
$SVUPC           030114
$SVUUO           030113
..CSC            000004  spd
..CSN            000003  spd
..MX1            002572' spd
..MX2            000001  spd
..PST            000003  spd
.CLOSE           003152' ext
.CONI            000117' ent
.CONO            000127' ent
.DATAI           000140' ent
```

```
AADDR        18     670#
ACSET      1456    1476     1483#
ACSET1     1470    1484#
```

```
AADDR      18     670#
ACSET    1456    1476     1483#
ACSET1   1470    1484#
ADRPE    1966
APR       378     962     1066     1237     1259
C20D     1831#
CADTBL   1334    1487     1573     1932#
CBASE      31     881      885     1337     1597     1928#
CBUF       50#   1708
CCWPN0   1729#   1741     1768     1772
CCWPN1   1737    1745#
CCWPNT     30    1703#
CCWPNX   1730    1769     1776#
CCWST    1554    1569#
CCWST1   1571    1577#
CHARG    1713    1818     1819     1826     1830     1838#
CHARG1   1711    1877     1878     1880     1881     1885     1890#
CHCNT    1935#
CHDATA     30    1959#
CHINO    1347#
CHINIT     29    1324#
CHINX    1341    1356#
CHK      1934#
CHKCSR     16     506#
CHKCSX    510     521      525#
CHP10    1434    1865#
CHP20    1504    1798#
CHPGM0   1817#   1821
CHPGM1   1818#
CISEL      50#
CLREBE    489     495#
CLREBU     16     487#
CLREBX    491     494#
CP10C    1873    1877#
CPOPJ    2434
CPOPJ1   2032
CPUNUM   2049#   2049     2065     2308
CPUTST   1997    2048#
CRAMPE    395     403      408      412      416      511      579      637      711     1081     1082     1095     1142
CSTWRD     50#   1961     1962     1963
CWBACK   1867    1874#
CZF      1449    1455#
CZFD     1459    1475#
DEBTIM     46#    966     1073
DEVADR   1599    1825     1879     2120#    2120     2265#    2265
DEVREL     35    2097#    2253     2469
DEVREQ     35    2225#
DEVRQX   2284#   2313     2318     2356
DGEPN0   2382#   2384
DGEPN1   2364    2379#
DGEPNT   1912    2168     2300     2362#
DGEPNX   2375    2390#
DIAGER   1603    1832     1883     1902#
```

```
DPRCX    2054    2074#
DSETUP     35    1989#
EADDR      18     669#
```

```
DPRCX     2054    2074#
DSETUP      35    1989#
EADDR       18     669#
EBUSPE     404     512     1083     1096     1156
EICHN     2151    2327     2340
EIFCN     2149    2325     2338
EISST     2345
ENABLE    2006    2443#
ERESET      17     375#
FFF         40    1466     1809     2026     2034     2052     2073     2469#
FILWDS    1440#   1440     1455     1467     1479     1500     1544     1546
FQUERR    1096    1159
GENC        29    1428#
GENGL     1430    1432     1439#
GETERR    2331    2344     2352#
GETLOG      30    1592#    1960
GRINDF    1492    1519#    1555
HAVECI      18    2106     2134     2202#    2240     2251     2279
HAVENI      18    2104     2136     2203#    2244     2252     2281
ICWA0      879     883     1336     1596     1939#
ICWA1     1940#
ICWA2     1941#
ICWA3     1942#
ICWA4     1943#
ICWA5     1944#
ICWA6     1945#
ICWA7     1946#
ICWPT     1636    1644#
IDELAY      25     889     1104     1105     1119#
INDIAG    2132    2161#    2161     2284     2291
INITP0     804#    817
INITPD      22     814#
INITPI      22     786#
INTAPR      24     963     1067     1124#
INTCON      24     961     1065     1123#
INTCSR      25    1070     1097     1127#
INTEND      25     808     1129#
INTNNN     957     973     1061     1107     1128#
INTNUM      24     807      888      950      956     1051     1060     1120#
INTPC       24     959     1063     1121#
INTTIM      24     967     1074     1125#
INTTYP      24     955     1059     1122#
INTUSE      24     969     1076     1126#
IPACLR      17     380      434#    1089     2113     2282
IPASR0     592#
IPASR1     580     603#
IPASR2     575     604#
IPASR3     573     605#
IPASRT      17     568#
IPASS0     724#
IPASS1     743     754#
IPASS2     712     771#
IPASS3     707     772#
```

```
IPASS4     705     773#
IPASST      17     699#
IPAST1     638     659#
```

```
IPASS4     705     773#
IPASST      17     699#
IPAST1     638     659#
IPAST2     633     660#
IPAST3     631     645     661#
IPASTP      17     626#
ISTOP       17     389#
ISTOPO     396     399#
ISTOPC     402     409     411     421#
ISTOPX     398     420     425#
JMPWD     1348    1572    1930#
LDCRAM      11     188#
LDCSR       11     145#    405     422     426     436     455     475     493     513     523     585     601     643
           651     717     732     745     750     764     768     830     847    1084    1091
LDEBUF      11     131#    766
LDRAR       11     167#    588     720
LOGARG    1600    1601    1604    1614#
LOGBUF      31    1606    1608    1645    1652    1675    1682    1685    1688    1933#   1935    1961
LOGPNT      30    1628#
LOSPT     1637    1651#
LOSPT1    1655#   1673
LSTCCW    1724#   1724    1736    1745    1749    1758    1761    1766
LSTDON    1496#
MAXWD     1927#
MBCN       135     149     171     192     226     241     267     286     306     326     346     366    1330    1484
          1595    1630    1802    2103    2105    2107    2108    2115    2118    2125    2127    2133    2135    2178
          2180    2232    2233    2239    2243    2250    2254    2260    2263    2270    2272    2278    2280
MBERR      401     408     509     579     637     711    1088    1095    1149
MCPUS     2055#   2055    2058
MONTYP    1431    1712    1992    2116    2146    2261    2322    2363    2420    2423
MPRUN      397     410     520     574     593     595     632     642     706     725     726
MVRERR    1096    1162
NIARG     2149    2151    2152    2198#   2325    2327    2328    2338    2340    2341    2345
NISEL       50#
NOSET     1987    2031#
NVIIN1     899     985#    989
NVIIN2     901     994#    998
NVIIN3     903    1003#   1007
NVIIN4     905    1012#   1016
NVIIN5     907    1021#   1025
NVIIN6     909    1030#   1034
NVIIN7     911    1039#   1043
NVIINC     992    1001    1010    1019    1028    1037    1046    1050#
NVJEN      990     999    1008    1017    1026    1035    1044    1062    1113    1115#
ODELA       51#   1106
P          236     246     456     476     494     516     602     603     604     658     659     660     770     771
           772    1338    1483    1497    1807    1816    1906    2225    2295    2355    2471
PCLEAR     421     435
PF500       42    1196    1207#   1239
PF501       42    1197    1208#   1241
PF502       42    1198    1209#   1243
PFAC0     1199    1210#   1236    1252    1260
PFAIL       42    1194    1211#   1257
```

```
PFAPR       42    1195    1206#   1237
PFCHK     1192    1236#
```

```
PFAPR       42     1195    1206#   1237
PFCHK     1192     1236#
PFCHKO    1247     1257#
PFCLR       41     1218#
PFDISP    1191     1205#   1223    1253
PFSET       41     1185#
PHYLST    1617#
PI         789      804     805     816     858     867     943     944     986     987     995     996    1004    1005
          1013     1014    1022    1023    1031    1032    1040    1041
PICLR      789      816
PIOFF      858      944     987     996    1005    1014    1023    1032    1041
PION       805
PISYOF      22      826     842     857#
PISYON      23      848     866#
PMODE       51#     413     417     423
PORTCI      50#    2107    2125    2133    2178    2232    2239    2270    2278
PORTNI      50#    2105    2127    2135    2180    2233    2243    2272    2280
PRCINI    2166     2178#   2298    2306    2311
PRSFLG     700#     700     742
PRVCHK    2000     2010#
RDCRAM      12      282#
RDCSR       12      236#    392     406     449     469     488     507     572     630     644     704     736     827
           843     1068
RDEBUF      12      222#    746
RDLAR       12      263#    652     756    1085    1092
REL         35     2163#
REL1      2153#
RELARG    2121     2162    2187#
RELCI     2126     2147    2161#
RELNI     2128     2146#   2354
REQ1        35     2292#
REQ2        35     2329#
REQARG    2192#    2266    2290
REQCI     2271     2290#   2323
REQCI2    2304     2315#
REQNI     2273     2322#
REQOK     2278#    2294    2347
RESQAV    1096     1165
RUNTME      46#     965    1072
SAC1      1903#    1903    1909    2164    2296    2302    2369    2379
SAVEBF     747#     747     765
SCLP      1335     1486    1496    [illegible]31#
SDATA       18      592     646    [illegible]68#    724     738     748     762     767
SELLAR     453      473     595    [illegible]50     726    1083    1090
SET2N      887      898#
SETEBE     450      457#
SETEBU      16      448#
SETLAE     470      477#
SETLAR      16      468#
SETVEC      23      874#
SINCYC     594      726
SKPCCW    1725#    1725    1738    1739    1746
SLAST       18      667#    755
```

```
SNEXT       18      586     654     666#    718     754     758
SNKMON    1993     2004#
```

```
SNEXT       18     586     654     666#    718     754     758
SNKMON    1993    2004#
START     2477    2478    2484
STRE      1538    1548    1554#
SWITT       46#
T10TAB    2382    2385    2395#   2405
T10TLN    2380    2405#
TCWPT     1638    1681#
TPSMON    1997#
TSTEBF     452     472     490     492     584     716     744     749     763
UDEBUG      50#    131     145     167     188     222     237     263     282     302     322     342     362     375
           389    1339    1429    1501    1592    1989    2031    2097    2226    2415    2431    2443
USER       377     379     596     727     786     814     823     839     857     866     875    1185    1218    1340
          1428    1503    1598    1709    1990    2098    2227    2416    2432    2444    2475
USRIOF      36    2431#
USRION      36    2010    2255    2415#
USTIM       46#    968    1075
VIINT      878     942#    958     979
VIIPI      943#    943     960     986     995    1004    1013    1022    1031    1040    1064
WCCHK     1525    1530#
WFIT      1531    1542#
WFIT1     1543    1545    1549#
ZI          58#
$EICHN    2151#   2327#   2340#
$EIFCN    2149#   2325#   2338#
$EISST    2345#
$SVUPC     946     978    1051    1112
$SVUUO     946     979    1051    1113
%EICHN    2151    2327    2340
%EIFCN    2149    2325    2338
%EISST    2345
..CSC     2149#   2149    2150    2150#   2151#   2151    2152    2152#   2325#   2325    2326    2326#   2327#   2327
          2328    2328#   2338#   2338    2339    2339#   2340#   2340    2341    2341#   2345#   2345    2346    2346#
..CSN     2149#   2149    2151#   2151    2325#   2325    2327#   2327    2338#   2338    2340#   2340    2345#   2345
..MX1     2148#   2148    2150#   2150    2152#   2324#   2324    2326#   2326    2328#   2337#   2337    2339#   2339
          2341#
..MX2     2148#   2149    2150#   2151    2324#   2325    2326#   2327    2337#   2338    2339#   2340
..PST     2149#   2149    2151#   2151    2325#   2325    2327#   2327    2338#   2338    2340#   2340    2345#   2345
.CLOSE      46#   2482
.CONI       10     302#
.CONO       10     322#
.DATAI      10     342#
.DATAO      10     362#
.DIAG     1602    1831    1832    1987#   1999    2005    2163    2292
.EICHN    2151    2327    2340
.EIFCN    2149    2325    2338
.EIGET    2324
.EIRCI    2337
.EIREL    2148
.EISOF    2346
.EISTA    2345
.EXIT       46#   2483
.FHSLF    2021    2370    2447
```

```
.INPNT      23    1135#
.INPNX    1141    1148    1155    1158    1161    1164    1170#
```

```
.INPNT      23      1135#
.INPNX    1141      1148      1155      1158      1161      1164      1170#
.INWD1      25       874       977      1098      1111      1135      1138      1170      1176#
.INWD2      25      1099      1144      1151      1177#
.JSYS     1986#     2004
.PIOFF      22       823#
.PION       22       839#      975      1109
.PRIOU    2020      2368
.UUO      1985#     1998
```

CI port>

```
CAXN      2346
EPCAP%    2451
```

```
EQ 2063   CAXN          2346                                                                                                                 SEQ 2076
          EPCAP%        2451
          ERJMP         2154    2330    2343    2449    2452
          ERSTR%        2023    2372
          FMSG          1911    2167    2299    2305    2310
          FMSGC         1905    1908    2018    2056    2059    2071    2165    2297    2470
          FMSGCD        2033    2072
          FMSGD         2312    2316
          GET            137     152     175     176     195     196     228     247     270     289     308     329     348     368
                         427     428     437     457     458     477     478     495     496     525     605     606     661     662
                         773     774     809     810     831     849     890     891     913     978     979    1112    1113    1171
                        1172    1200    1225    1332    1356    1357    1498    1499    1500    1577    1609    1610    1639    1640
                        1777    1778    1833    1834    1884    1885    1914    1915    1967    1999    2005    2012    2074    2075
                        2140    2141    2254    2285    2286    2390    2391    2426    2427    2453    2454
          GETTAB        2051
          GO             380     392     405     406     422     426     436     449     455     469     475     488     493     507
                         512     513     523     572     585     588     601     630     643     644     651     652     704     717
                         720     732     736     745     746     750     756     764     766     768     826     827     830     842
                         843     847     848     887     965     975    1068    1072    1082    1084    1085    1089    1091    1092
                        1097    1106    1109    1434    1466    1492    1504    1554    1599    1603    1636    1637    1638    1809
                        1832    1883    1912    1960    1987    1997    2006    2010    2011    2026    2034    2052    2073    2113
                        2126    2128    2166    2168    2253    2255    2282    2298    2300    2306    2311    2469    2482    2483
          HALT          2477
          JEN            979    1113    1262
          JRSTF         2141    2434
          JSYS          1986
          LOAD          2345
          MOVX          2148    2150    2152    2324    2326    2328    2337    2339    2341
          NI%           2153    2329    2342
          OP%%CA        2346
          PCRL          1747    1776
          PCRLF         1904    1913    2169    2301    2474
          PDEC          1759
          PNT1          1633
          PNT6F         2473
          PNTADR        1762
NI port>  PNTCI         2179    2181    2182
          PNTCIF        2307    2309
          PNTDEC        1686
          PNTHW         1646    1683    2388
          PNTMSF        1905    1908    1911    2019    2033    2057    2059    2071    2072    2165    2167    2297    2299    2305
                        2310    2312    2317    2470
          PNTMSG         415     419     424     514     524    1140    1143    1147    1150    1154    1157    1160    1163    1166
                        1629    1635    1644    1651    1656    1657    1658    1659    1660    1666    1667    1668    1669    1670
                        1674    1681    1684    1687    1717    1718    1719    1740    1750    1751    1752    1753    1754    1755
                        1756    1757    1760    2154    2330    2343    2348    2353    2386
          PNTOCC        1748
          PNTOCF        1907    1910
          PNTOCS        1146    1153    1676    1689
          PUT            133     147     169     170     190     191     224     239     265     284     304     324     344     364
                         391     392     434     448     449     468     469     487     488     506     568     569     626     627
                         699     700     788     806     825     841     877     878     898     945     946     988     997    1006
                        1015    1024    1033    1042    1050    1051    1137    1138    1187    1220    1324    1325    1329    1440

SEQ 2064                1441    1442    1443    1569    1594    1595    1628    1629    1703    1704    1798    1799    1865    1866   SEQ 2077
                        1902    1903    1959    1998    2004    2048    2049    2112    2113    2250    2259    2260    2362    2363
```

```
                1441    1442    1443    1569    1594    1595    1628    1629    1703    1704    1798    1799    1865    1866
                1902    1903    1959    1998    2004    2048    2049    2112    2113    2250    2259    2260    2362    2363
                2418    2419    2446    2447
RGET             175     195     427     457     477     495     605     661     773     809     890     978    1112    1171
                1356    1498    1499    1609    1639    1777    1833    1884    1914    2074    2140    2285    2390    2426
                2453
RPCAP%          2448
RPUT             169     190     391     448     468     487     568     626     699     877     945    1050    1137    1324
                1441    1442    1594    1628    1703    1798    1865    1902    2048    2112    2259    2362    2418    2446
RTN              132     138     146     153     168     176     189     196     223     229     238     248     264     271
                 283     290     303     309     323     330     343     349     363     369     376     381     390     428
                 438     458     478     496     515     526     606     662     774     787     810     815     824     832
                 840     850     859     868     876     891     914    1087    1094    1100    1136    1172    1186    1201
                1219    1226    1357    1435    1502    1505    1556    1582    1593    1605    1610    1640    1647    1677
                1690    1778    1834    1886    1915    1968    1991    2014    2075    2099    2108    2142    2157    2170
                2174    2183    2228    2235    2241    2245    2286    2391    2417    2427    2433    2445    2454
SETUUO          2070
STOR            2149    2151    2325    2327    2338    2340
TMSG            1634    1656    1657    1658    1659    1660    1666    1667    1668    1669    1670    1674    1684    1687
                1750    1751    1752    1753    1754    1755    1756    1757    1760
TMSGC           1143    1150    1629    1644    1651    1681    1717    1718    1719    1740    2330    2343    2348
TMSGCD           415     419     424     514     524    1140    1157    1160    1163    1166    2154    2352
TMSGD           1147    1154
TRPSET          2421
TTIDEC          2060
USRIO%          2424
%EICHN          2151    2327    2340
%EIFCN          2149    2325    2338
%EISST          2345
```

```
DFPTAU  MACRO %53B(1242) 10:16  1-Jul-85 Page 1
DFPTAU  MAC       16-May-85 17:43          UTILITY Routines Module
```

```
        TITLE   DFPTAU
        SUBTTL  UTILITY Routines Module

        SALL
        .DIREC  FLBLST          ;First line binary only in listing

        SEARCH  DFPTA,MONSYM

; Typeout/typin routines

        ENTRY   CLRBUF,TTYYES,PNTDEV
        ENTRY   PPDEC,PPDECF,PSDN,ALTCHH,CONVSX,.PNTOC

; Clock handling routines/variables

        ENTRY   STCLOK,MONTIM,PTIME,RUNTME,PSTAMP
        INTERN  CURTIM,MAXTIM,DEBTIM,USTIM,FORPNT,LCHAR

; TOPS-10/20 miscellaneous routines

        ENTRY   .CLOSE,.RESET,.EXIT

; File handling routines

        ENTRY   FINIT,FGETW,FINCMD,FOARG
        INTERN  UNAME

; Data pattern routines

        ENTRY   BUFGEN,BUFCOM,PATPNT,PATBUF
        INTERN  WRDERR

; Switch handling routines

        ENTRY   .SWCHP,SWITT,.ISWT,SWCHPT,SWCOM
        INTERN  .RSWIT,.LSWIT,SWRGT

; Miscellaneous routines

        ENTRY   PARSER,.COMM,.SARG,.OARG,.DARG,CHKARG,LASARG,DECYN,FIOFF
        INTERN  .CGOT,ARGUM,ARGFLG

; Scope routine

        ENTRY   SCOPE1
        INTERN  SCOSW,SCOOFF

; EXTERN's located in DFPTA1.MAC

        EXTERN  UDEBUG,MDEBUG,BUFF,CADDR,CWORDL,CWORDR,UUT,SPREP2
        EXTERN  INDLIS,TSTCLS,TSTNUM,TSTSUB,TSTPC,TSTPNT,TSTREP,TSTOFF
        EXTERN  FINPUT,FINECH,TAKFIL,TPAT,PORTCI,PORTNI,NISEL,CISEL,PORSEL
        EXTERN  ALTF,TSTINH
```

```

                                     ; EXTERN's located in DFPTAI.MAC (Port Device Handling Routines module)

                                             EXTERN  RDCSR,GETLOG,LOGPNT,FFF


                                     ;#*******************************************************************
                                     ; ZU - Address for use in DDT
                                     ;#*******************************************************************

 000000'                             ZU:                                  ; address of 00000'

```

```
                                                    SUBTTL  Printout/Typin Routines

                                             ;#*******************************************************************
                                             ;* CONVSX - Convert sixbit to octal
                                             ;
                                             ; Argument:      AC1 - Contains a word of sixbit (which may or may not
                                             ;                      be an octal number).
                                             ;
                                             ; Return:        +1 - Error - not a valid octal number
                                             ;                +2 - Ok - number converted ok - in AC1
                                             ;#*******************************************************************

000000' 261 17 0 00 000000                   CONVSX: RPUT     (0,2)                      ; save AC's
000002' 202 01 0 00 006114'                          MOVEM    1,CONVSS#                  ; save argument
000003' 200 01 0 00 004063'                          MOVE     1,[POINT 6,CONVSS]         ; set to get sixbit char
000004' 202 01 0 00 000027'                          MOVEM    1,CONVSP                   ; save the pointer
000005' 400 02 0 00 000000                           SETZ     2,                         ; clear result
000006' 134 01 0 00 000027'                  CONVSO: ILDB     1,CONVSP                   ; get a char
000007' 322 01 0 00 000022'                          JUMPE    1,CONVS1                   ; zero - all done - exit
000010' 301 01 0 00 000020                           CAIL     1,20                       ; see if its in octal range
000011' 303 01 0 00 000027                           CAILE    1,27
000012' 254 00 0 00 000017'                          JRST     CONVSE                     ; no - error
000013' 405 01 0 00 000007                           ANDI     1,7                        ; yes - save only digit
000014' 242 02 0 00 000003                           LSH      2,3                        ; multiply by 8
000015' 272 01 0 00 000002                           ADDM     1,2                        ; insert in number
000016' 254 00 0 00 000006'                          JRST     CONVSO                     ; get next digit

                                             ; Error

000017' 262 17 0 00 000002                   CONVSE: RGET     (2,0)                      ; restore AC's
000021' 263 17 0 00 000000                           RTN                                 ; return +1

                                             ; Done - put result into AC1

000022' 200 01 0 00 000002                   CONVS1: MOVE     1,2
000023' 262 17 0 00 000002                           RGET     (2,0)                      ; restore AC's
000025' 350 00 0 17 000000                           AOS      (P)                        ; set up return +2
000026' 263 17 0 00 000000                           RTN                                 ; exit

000027' 000000  000000                       CONVSP: 0                                   ; number to convert
000030' 000000  000000                               0                                   ; trailing zeros
```

```

                                         ;#*********************************************************************
                                         ;* ALTCHH - Handle altmode intercept
                                         ;
                                         ; If an altmode has been typed in response to the question asked
                                         ; immediately before this call - exit through ALTMGO.
                                         ;
                                         ; If an altmode has not been typed - exit to the address specified
                                         ; by the right half of the UUO (in $SVUUO).
                                         ;#*********************************************************************

000031' 261 17 0 00 000000               ALTCHH: PUT     0                       ; save AC0
000032' 200 00 0 00 030232                       MOVE    $CHRIN                  ; get last character typed
000033' 306 00 0 00 000033                       CAIN    33                      ; altmode?
000034' 254 00 0 00 000042'                      JRST    ALTCHX                  ; yes - handle
000035' 302 00 0 00 000175                       CAIE    175                     ; altmode?
000036' 306 00 0 00 000176                       CAIN    176                     ; altmode?
000037' 254 00 0 00 000042'                      JRST    ALTCHX                  ; yes - handle
000040' 262 17 0 00 000000                       GET     0                       ; no - restore AC0
000041' 254 00 1 00 030113                       JRST    @$SVUUO                 ; exit

000042' 262 17 0 00 000000               ALTCHX: GET     0                       ; restore AC0
000043' 262 17 0 00 006137'                      GET     XX#                     ; take off return address
000044' 254 00 1 00 030063                       JRST    @ALTMGO                 ; go to intercept address



                                         ;#*********************************************************************
                                         ;* CLRBUF - Clear input buffer
                                         ;#*********************************************************************

000045' 336 00 0 00 030037               CLRBUF: SKIPN   USER                    ; user mode?
000046' 263 17 0 00 000000                       RTN                             ; no - return
000047' 261 17 0 00 000000                       RPUT    (0,1)                   ; save some AC's
000051' 201 01 0 00 000100                       MOVEI   1,100                   ; set up TOPS10 argument
000052' 332 00 0 00 030516                       SKIPE   MONTYP                  ; TOPS-20?
000053' 104 00 0 00 000100                       CFIBF                           ; yes - clear TTY input buffer
000054' 336 00 0 00 030516                       SKIPN   MONTYP                  ; TOPS-10?
000055' 051 11 0 00 000000                       CLRBFI                          ; yes - clear input buffer
000056' 262 17 0 00 000001                       RGET    (1,0)                   ; restore AC's
000060' 263 17 0 00 000000                       RTN                             ; return
```

```

                                  ;#*********************************************************************
                                  ;* TTYYES - Answer yes or no question
                                  ;
                                  ; Calling sequence:  MOVE [FMSG  <Question?>]
                                  ;                    GO   TTYYES
                                  ;                    Return+1 ==> No
                                  ;                    Return+2 ==> Yes
                                  ;#*********************************************************************

000061' 261 17 0 00 000000        TTYYES: RPUT    (0,1,2)                        ; save AC's
000064' 202 00 0 00 000002                MOVEM   2                              ; get printout in AC2
000065' 256 00 0 00 000002        TTYY0:  XCT     2                              ; print question
000066' 037 01 0 00 004064'               FMSG    < (Y or N) - >                 ; print Y or N section
000067' 037 10 0 00 000003        TTYY1:  TTSIXB                                 ; get it
                                          ALTCHK  [SKIPE  $TWCNT                 ; did a timeout occur?
                                                   JRST   TTYY0                  ; no - error - reask question
000070' 007 00 0 00 004067'                        JRST   TTYY1]                 ; yes - just keep looking for answer
000071' 404 00 0 00 004072'               AND     [770000,,0]                    ; mask out all but first character
000072' 241 00 0 00 000006                ROT     6                              ; right justify byte
000073' 302 00 0 00 000071                CAIE    71                             ; Y?
000074' 306 00 0 00 000056                CAIN    56                             ; N?
000075' 334 00 0 00 000000                SKIPA                                  ; yes - continue
000076' 254 00 0 00 000065'               JRST    TTYY0                          ; neither - reask question
000077' 306 00 0 00 000071                CAIN    71                             ; Y?
000100' 350 00 0 17 777775                AOS     -3(P)                          ; increment return
000101' 262 17 0 00 000002                RGET    (2,1,0)                        ; restore AC's
000104' 263 17 0 00 000000                RTN                                    ; return
```

```


                                    ;#***********************************************************************
                                    ;* PSDN - Prints signed decimal numbers
                                    ;
                                    ; PSDN prints the contents of AC0 as a signed decimal number followed
                                    ; by a period.  Leading zeros are suppressed.
                                    ;#***********************************************************************

000105' 261 17 0 00 000000          PSDN:   PUT
                                            JUMPL   [TMSG   <->
                                                     MOVN   (P)
000106' 321 00 0 00 004074'                          JRST   .+1]
000107' 037 15 0 00 000000                  PNTDEC                          ; print the number
000110' 037 00 0 00 004077'                 TMSG    <.>                     ; follow with a period
000111' 262 17 0 00 000000                  GET
000112' 263 17 0 00 000000                  RTN

000113' 261 17 0 00 000000          PSDNF:  PUT
                                            JUMPL   [FMSG   <->
                                                     MOVN   (P)
000114' 321 00 0 00 004100'                          JRST   .+1]
000115' 037 15 0 00 000001                  PNTDCF
000116' 037 01 0 00 000056                  PNTCIF  "."
000117' 262 17 0 00 000000                  GET
000120' 263 17 0 00 000000                  RTN
```

```


                                          ;#*****************************************************************
                                          ;* .PNTOC - Print n digit octal number with leading zeros suppressed
                                          ;#*****************************************************************

000121' 261 17 0 00 000000                .PNTOC: RPUT    (0,1,2,3,4,$SVUUO,$SVUPC)
000130' 135 01 0 00 004103'                       LDB     1,[POINT 23,$SVUUO,35]  ; get address portion
000131' 661 01 0 00 200040                        TLO     1,200040                ; make into a "MOVE 1," instruction
000132' 261 17 0 00 000001                        PUT     1                       ; put on stack
000133' 200 01 0 17 777772                        MOVE    1,-6(P)                 ; restore AC1
000134' 256 00 0 17 000000                        XCT     (P)                     ; get number into AC1
000135' 262 17 0 00 006137'                       GET     XX#                     ; pop off temporary stack location
000136' 214 02 0 00 000001                        MOVM    2,1                     ; get absolute value
000137' 243 02 0 00 000141'                       JFFO    2,.+2                   ; find out where number starts
000140' 201 03 0 00 000043                        MOVEI   3,^D35                  ; assume at least one bit set
000141' 231 03 0 00 000003                        IDIVI   3,3                     ; calculate # of leading 0's there are
000142' 135 02 0 00 004104'                       LDB     2,[POINT 4,$SVUUO,12]   ; get AC field
000143' 270 02 0 00 000003                        ADD     2,3                     ; calculate number of spaces to
000144' 275 02 0 00 000014                        SUBI    2,^D12                  ;   print
000145' 335 00 0 00 000001                        SKIPGE  1                       ; number is actually negative?
000146' 370 00 0 00 000002                        SOS     2                       ; yes - allow for minus sign
000147' 333 00 0 00 000002                        SKIPLE  2                       ; number of spaces still 1..n?
                                                  XCT     [JFCL                   ; yes - print leading spaces
                                                           TMSG   < >             ; 1
                                                           TMSG   <  >            ; 2
                                                           TMSG   <   >           ; 3
                                                           TMSG   <    >          ; 4
                                                           TMSG   <     >         ; 5
                                                           TMSG   <      >        ; 6
                                                           TMSG   <       >       ; 7
                                                           TMSG   <        >      ; 8
                                                           TMSG   <         >     ; 9
                                                           TMSG   <          >    ; 10
000150' 256 00 0 02 004131'                                TMSG   <           >](2)
000151' 200 00 0 00 000001                        MOVE    1                       ; get number
000152' 037 16 0 00 000003                        PNTOCS                          ; print it
000153' 262 17 0 00 030114                        RGET    ($SVUPC,$SVUUO,4,3,2,1,0)
000162' 263 17 0 00 000000                        RTN                             ; exit
```

```

                                    ;#*********************************************************************
                                    ;* PPDEC - Print decimal number in a field of size x.
                                    ;
                                    ; Call is:  PDEC X,ADR
                                    ;#*********************************************************************

000163' 261 17 0 00 000000          PPDEC:  RPUT    (0,1,$SVUUO,$SVUPC)     ; save ACs
000167' 200 00 1 00 030113                  MOVE    @$SVUUO                 ; get number to print
000170' 135 01 0 00 004104'                 LDB     1,[POINT 4,$SVUUO,12]   ; get size field
000171' 322 00 0 00 000202'                 JUMPE   PDECSS                  ; special print
000172' 250 01 0 00 030264          PDECO:  EXCH    1,RADLSC                ; put in position
000173' 037 16 0 00 000000                  PNTDS                           ; no - print the number
000174' 202 01 0 00 030264                  MOVEM   1,RADLSC                ; restore old field size
000175' 262 17 0 00 030114                  RGET    ($SVUPC,$SVUUO,1,0)     ; restore ACs
000201' 263 17 0 00 000000                  RTN                             ; exit

000202' 322 01 0 00 000172'         PDECSS: JUMPE   1,PDECO                 ; make sure length field is specified
                                            XCT     [TMSG   <0>             ; print a bunch of spaces
                                                     TMSG   <0>
                                                     TMSG   < 0>
                                                     TMSG   <  0>
                                                     TMSG   <   0>
                                                     TMSG   <    0>
                                                     TMSG   <     0>
                                                     TMSG   <      0>
                                                     TMSG   <       0>
                                                     TMSG   <        0>
                                                     TMSG   <         0>
                                                     TMSG   <          0>
                                                     TMSG   <           0>
                                                     TMSG   <            0>
000203' 256 00 0 01 004206'                          TMSG   <             0>](1)
000204' 262 17 0 00 030114                  RGET    ($SVUPC,$SVUUO,1,0)     ; restore ACs
000210' 263 17 0 00 000000                  RTN                             ; exit
```

```

                                ;#********************************************************************
                                ;* PPDECF - Force print decimal number in a field of size x.
                                ;
                                ; Call is:  PDECF X,ADR
                                ;#********************************************************************

000211' 261 17 0 00 000000      PPDECF: RPUT    (0,1,$SVUUO,$SVUPC)     ; save ACs
000215' 200 00 1 00 030113              MOVE    @$SVUUO                 ; get number to print
000216' 135 01 0 00 004104'             LDB     1,[POINT 4,$SVUUO,12]   ; get size field
000217' 322 00 0 00 000230'             JUMPE   PDECFS                  ; special print
000220' 250 01 0 00 030264      PDECFO: EXCH    1,RADLSC                ; put in position
000221' 037 16 0 00 000001              PNTDSF                          ; no - print the number
000222' 202 01 0 00 030264              MOVEM   1,RADLSC                ; restore old field size
000223' 262 17 0 00 030114              RGET    ($SVUPC,$SVUUO,1,0)     ; restore ACs
000227' 263 17 0 00 000000              RTN                             ; exit

000230' 322 01 0 00 000220'     PDECFS: JUMPE   1,PDECFO                ; make sure length field is specified
                                        XCT     [FMSG   <0>             ; print a bunch of spaces
                                                 FMSG   <0>
                                                 FMSG   < 0>
                                                 FMSG   <  0>
                                                 FMSG   <   0>
                                                 FMSG   <    0>
                                                 FMSG   <     0>
                                                 FMSG   <      0>
                                                 FMSG   <       0>
                                                 FMSG   <        0>
                                                 FMSG   <         0>
                                                 FMSG   <          0>
                                                 FMSG   <           0>
                                                 FMSG   <            0>
                                                 FMSG   <             0>
000231' 256 00 0 01 004226'                      FMSG   <              0>](1)
000232' 262 17 0 00 030114              RGET    ($SVUPC,$SVUUO,1,0)     ; restore ACs
000233' 263 17 0 00 000000              RTN                             ; exit

```

```
                                                SUBTTL  Clock Handling Routines

                                        ;#*********************************************************************
                                        ;* STCLOK - Initialize program runtime clock
                                        ;
                                        ; This routine initializes the clock for the program.  It is used in
                                        ; conjunction with routine 'RUNTME'.
                                        ;
                                        ;   In exec mode this routine resets, restarts and  reads  the current
                                        ; microsecond time clock into  memory  for  reference  by  the runtime
                                        ; calculator 'RUNTME'.  The clock is set up so  it  causes  a hardware
                                        ; interrupt, it merely free runs.
                                        ;
                                        ;   In user mode the program run time is obtained in milliseconds with
                                        ; a  monitor  call, converted  to  microseconds  and stored for future
                                        ; reference by the runtime calculator 'RUNTME'.
                                        ;
                                        ;   This routine may be called anytime to reset the program runtime.
                                        ;
                                        ;   Call Sequence:      GO  STCLOK          ; calls the routine
                                        ;                       RTN-1               ; always returns +1
                                        ;#*********************************************************************

000237' 261 17 0 00 000000              STCLOK: RPUT    (0,1)                   ; save AC's
000241' 332 00 0 00 030037                      SKIPE   USER                    ; in user mode?
000242' 254 00 0 00 000251'                     JRST    STCU                    ; yes
000243' 7 024 20 0 00 420000                    CONO    24,1B18!1B22            ; stop and reset the clock
000244' 402 00 0 00 000510                      SETZM   510                     ; clear EPT locations
000245' 402 00 0 00 000511                      SETZM   511                     ;   for timer data
000246' 7 024 20 0 00 002000                    CONO    24,1B25                 ; turns the meter on
000247' 7 020 04 0 00 000260'                   DATAI   20,TMRST1               ; save a double word
000250' 254 00 0 00 000255'                     JRST    STCX                    ; done - exit
000251' 476 00 0 00 006125'             STCU:   SETOM   INITM#                  ; invalidate initial time
000252' 260 17 0 00 000262'                     GO      MONTIM                  ; get monitor time in microseconds
000253' 255 00 0 00 000000                      JFCL                            ;   in meter format
000254' 124 01 0 00 000260'                     DMOVEM  1,TMRST1                ; save a double word
000255' 262 17 0 00 000001              STCX:   RGET    (1,0)                   ; restore
000257' 263 17 0 00 000000                      RTN                             ; exit

000260'                                 TMRST1: BLOCK   2                       ; 2 word initial time storage
```

```
                                        ;#***************************************************************
                                        ;* MONTIM - Read program runtime from monitor
                                        ;
                                        ;    Here is a subroutine called by both STCLOK and RUNTME.  It reads
                                        ; current program  runtime in milliseconds from the monitor, converts
                                        ; the time to microseconds and positions the microsecond count in AC2
                                        ; and AC1 in the same 59 bit format you get when you read  the  meter
                                        ; board microsecond clock.  The microsecond count is then returned in
                                        ; AC1 and AC2 .....
                                        ;
                                        ; Call:  GO      MONTIM             ; call the routine
                                        ;        RTN+1                      ; microsecond time in AC1 and AC2
                                        ;#***************************************************************

000262' 261 17 0 00 000003              MONTIM: PUT     3                           ; save AC3 (RUNTM JSYS destroys it)
000263' 332 00 0 00 030516                      SKIPE   MONTYP                      ; TOPS10?
000264' 254 00 0 00 000276'                     JRST    MONTI0                      ; no - go handle TOPS20
000265' 200 01 0 00 004246'                     MOVE    1,[136,,11]                 ; GETTAB item for system uptime in jiffies
000266' 047 01 0 00 000041                      GETTAB  1,                          ; ask the monitor
000267' 400 01 0 00 000000                      SETZ    1,                          ; not likely it will fail
000270' 221 01 0 00 001750                      IMULI   1,^D1000                    ; convert to milli-jiffies
000271' 336 00 0 00 030505                      SKIPN   CYCL60                      ; then to milliseconds
000272' 231 01 0 00 000074                      IDIVI   1,^D60                      ; if 60 Hz power frequency
000273' 332 00 0 00 030505                      SKIPE   CYCL60                      ; but if European frequency
000274' 231 01 0 00 000062                      IDIVI   1,^D50                      ; use different divisor
000275' 254 00 0 00 000301'                     JRST    MONTI1
000276' 201 01 0 00 400000              MONTI0: MOVEI   1,.FHSLF                    ; get process handle
000277' 104 00 0 00 000014                      TIME                                ; does a TIME JSYS
000300' 320 17 0 00 004247'                     ERCAL   [RTN]                       ; error not likely
000301' 335 00 0 00 006125'             MONTI1: SKIPGE  INITM#                      ; initial time valid?
000302' 202 01 0 00 006125'                     MOVEM   1,INITM#                    ; no - set it up
000303' 274 01 0 00 006125'                     SUB     1,INITM#                    ; subtract initial time
000304' 400 02 0 00 000000                      SETZ    2,                          ; clear AC2
000305' 225 01 0 00 001750                      MULI    1,^D1000                    ; converts ms to us
000306' 244 01 0 00 000014                      ASHC    1,^D12                      ; put data in proper microsecond format
000307' 262 17 0 00 000003                      GET     3                           ; restore AC3
000310' 263 17 0 00 000000                      RTN                                 ;   and exit



                                        ;#***************************************************************
                                        ;* PTIME - Print time of day
                                        ;#***************************************************************

000311' 261 17 0 00 000003              PTIME:  PUT     3                           ; save AC3
000312' 200 03 0 00 004131'                     MOVE    3,[JFCL]                    ; non-zero for RUNTME
000313' 402 00 0 00 006121'                     SETZM   FORPNT#                     ; clear forced print flag
000314' 260 17 0 00 000325'                     GO      RUNTME                      ; go print out time
000315' 262 17 0 00 000003                      GET     3                           ; restore AC3
000316' 263 17 0 00 000000                      RTN                                 ; return

000317' 261 17 0 00 000003              PTIMEF: PUT     3                           ; save AC3
000320' 200 03 0 00 004131'                     MOVE    3,[JFCL]                    ; non-zero for RUNTME
000321' 476 00 0 00 006121'                     SETOM   FORPNT                      ; set forced print flag
000322' 260 17 0 00 000325'                     GO      RUNTME                      ; go print out time
```

```
000323' 262 17 0 00 000003              GET     3                       ; restore AC3
000324' 263 17 0 00 000000              RTN                             ; return
```

```
                                ;#*********************************************************************
                                ;* RUNTME - Calculate and print program runtime
                                ;
                                ;    This reports the elapsed time (exec  or  user  mode) since the last
                                ; call to the initialize routine "STCLOK". The runtime is printed on the
                                ; current line and no line feed is  printed  at  the end of the message.
                                ; It is printed as  " 1h 20m 48s"
                                ;
                                ;    In exec mode the time is calculated by using the microsecond  clock
                                ; that was initialized in the call to "STCLOK".  In user mode, "RUNTIME"
                                ; monitor calls are used and the difference  is calculated from the last
                                ; call to "STCLOK".
                                ;
                                ;    Arguments:  AC3 - contains message to print - if zero then the time
                                ; is  only  calculated  and  the  routine returns (in TMSG format).  The
                                ; current time is always placed in CURTIM (in seconds).
                                ;
                                ;    Call sequence:     GO  RUNTME        ; call the routine
                                ;                       RTN+1             ; always returns +1
                                ;#*********************************************************************

000325' 261 17 0 00 000000      RUNTME: RPUT    (0,1,2,3)               ; save AC's

                                ; Get the microsecond count

000331' 332 00 0 00 030037              SKIPE   USER                    ; in user mode?
000332' 254 00 0 00 000335'             JRST    RUNTMO                  ; yes
000333' 7 020 04 0 00 000415'           DATAI   20,TMRUN1               ; exec mode. get dble word time
000334' 254 00 0 00 000337'             JRST    RUNTM1                  ; go to common code
000335' 260 17 0 00 000262'     RUNTMO: GO      MONTIM                  ; get time from mon calls to AC1,AC2
000336' 124 01 0 00 000415'             DMOVEM  1,TMRUN1                ; save microsecond count in double word

                                ; Calculate time

000337' 120 00 0 00 000415'     RUNTM1: DMOVE   TMRUN1                  ; get current run time
000340' 115 00 0 00 000260'             DSUB    TMRST1                  ; calculates actual run time
000341' 244 00 0 00 777764              ASHC    -^D12                   ; right justify
000342' 234 00 0 00 004250'             DIV     [^D1000]                ; convert us to ms
000343' 202 00 0 00 000421'             MOVEM   DEBTIM                  ; save in debug time location
000344' 202 01 0 00 000422'             MOVEM   1,USTIM                 ; save microseconds also
000345' 230 00 0 00 004250'             IDIV    [^D1000]                ; convert ms to seconds
000346' 202 00 0 00 000417'             MOVEM   CURTIM                  ; save in current time location

                                ; Exit if not printing

000347' 336 00 0 00 000003              SKIPN   3                       ; message to print?
                                        JRST    [RGET   (3,2,1,0)       ; no - restore AC's and
000350' 254 00 0 00 004251'                      RTN]                   ;   return
```

```


                                        ; Print the time

000351' 256 00 0 00 000003                      XCT     3                       ; print the message
000352' 231 00 0 00 000074                      IDIVI   ^D60                    ; compute minutes, seconds
000353' 202 01 0 00 000003                      MOVEM   1,3                     ; save seconds
000354' 231 00 0 00 000074                      IDIVI   ^D60                    ; compute hours, minutes
000355' 322 00 0 00 000366'                     JUMPE   RUNTM2                  ; print hours if not zero

                                        ; Print hours

000356' 332 00 0 00 006121'                     SKIPE   FORPNT                  ; forced print?
000357' 037 15 0 00 000001                      PNTDCF                          ; yes - print hours
000360' 336 00 0 00 006121'                     SKIPN   FORPNT                  ; forced print?
000361' 037 15 0 00 000000                      PNTDEC                          ; no - print hours

000362' 332 00 0 00 006121'                     SKIPE   FORPNT                  ; forced print?
000363' 037 01 0 00 004256'                     FMSG    <h >                    ; yes
000364' 336 00 0 00 006121'                     SKIPN   FORPNT                  ; forced print?
000365' 037 00 0 00 004256'                     TMSG    <h >                    ; no

                                        ; Print minutes

000366' 200 00 0 00 000001              RUNTM2: MOVE    1
000367' 332 00 0 00 006121'                     SKIPE   FORPNT                  ; forced print?
000370' 037 15 0 00 000001                      PNTDCF                          ; yes - print minutes
000371' 336 00 0 00 006121'                     SKIPN   FORPNT                  ; forced print?
000372' 037 15 0 00 000000                      PNTDEC                          ; no - print minutes
000373' 332 00 0 00 006121'                     SKIPE   FORPNT                  ; forced print?
000374' 037 01 0 00 004257'                     FMSG    <m >                    ; yes
000375' 336 00 0 00 006121'                     SKIPN   FORPNT                  ; forced print?
000376' 037 00 0 00 004257'                     TMSG    <m >                    ; no

                                        ; Print seconds

000377' 200 00 0 00 000003                      MOVE    3
000400' 332 00 0 00 006121'                     SKIPE   FORPNT                  ; forced print?
000401' 037 15 0 00 000001                      PNTDCF                          ; yes - print seconds
000402' 336 00 0 00 006121'                     SKIPN   FORPNT                  ; forced print?
000403' 037 15 0 00 000000                      PNTDEC                          ; no - print seconds
000404' 332 00 0 00 006121'                     SKIPE   FORPNT                  ; forced print?
000405' 037 01 0 00 004260'                     FMSG    <s >
000406' 336 00 0 00 006121'                     SKIPN   FORPNT                  ; forced print?
000407' 037 00 0 00 004261'                     TMSG    <s>

000410' 262 17 0 00 000003              RUNTMX: RGET    (3,2,1,0)               ; restore AC's
000414' 263 17 0 00 000000                      RTN                             ;   and exit
```

```
                                         ; Timer routine storage

000415'                                  TMRUN1: BLOCK   2               ; microsecond runtime saved here...

000417' 000000  000000                   CURTIM: 0                       ; current time in seconds
000420' 000000  000000                   MAXTIM: 0                       ; maximum wait time
000421' 000000  000000                   DEBTIM: 0                       ; current time in milliseconds
000422' 000000  000000                   USTIM:  0                       ; current usecs (additional to DEBTIM)


                                         ;#*********************************************************************
                                         ;* PSTAMP - Check for an 'S' typed.
                                         ;#*********************************************************************

000423' 261 17 0 00 000000               PSTAMP: RPUT    (0,1,2,3,4)     ; save AC's
000430' 200 00 0 00 006126'                      MOVE    LCHAR#          ; get last character typed
000431' 306 00 0 00 000123                       CAIN    123             ; was it an upper case-s?
000432' 254 00 0 00 000437'                      JRST    PSTAM0          ; yes - print current statistics
000433' 306 00 0 00 000163                       CAIN    163             ; lower case-s?
000434' 254 00 0 00 000437'                      JRST    PSTAM0          ; yes - print current statistics
000435' 302 00 0 00 000023                       CAIE    023             ; control-s?
000436' 254 00 0 00 000465'                      JRST    PSTAMX          ; no - exit

                                         ; Print runtime data

000437' 037 01 0 00 004262'              PSTAM0: FMSGC   <** >
000440' 260 17 0 00 004045'                      GO      PNTDEV          ; print device name
000441' 037 01 0 00 004264'                      FMSG    < Test >
000442' 200 01 0 00 000000*                      MOVE    1,TSTCLS        ; get test class
000443' 200 00 0 01 000000*                      MOVE    INDLIS(1)       ; get class description
000444' 037 01 0 00 000002                       PNTSXF                  ; print it
000445' 200 00 0 00 000000*                      MOVE    TSTNUM          ; get test number
000446' 037 17 0 00 000003                       PNTOCF                  ; print it
000447' 200 00 0 00 000000*                      MOVE    TSTSUB          ; get subtest number
000450' 322 00 0 00 000453'                      JUMPE   .+3             ; zero?  yes - continue
000451' 037 01 0 00 004073'                      FMSG    <->             ; no - print it
000452' 037 17 0 00 000003                       PNTOCF
000453' 037 01 0 00 004266'                      FMSG    < at >
000454' 260 17 0 00 000317'                      GO      PTIMEF          ; print current time
000455' 037 01 0 00 004267'                      FMSG    <in Pass >
000456' 260 17 0 00 001571'                      GO      SWITT           ; Get the switches
000457' 607 00 0 00 000010                       TLNN    OPRSEL          ; OPRSEL switch on?
000460' 334 01 0 00 030047                       SKIPA   1,PASCNT        ; SUBRTN is counting passes
000461' 200 01 0 00 000000*                      MOVE    1,TSTREP        ; I am counting the passes
000462' 201 00 0 01 000001                       MOVEI   1(1)            ; Load the pass ccunt
000463' 037 15 0 00 000001                       PNTDCF                  ; print it
000464' 037 01 0 00 004271'                      FMSGD   <. **>

000465' 402 00 0 00 006126'              PSTAMX: SETZM   LCHAR           ; clear last character typed location
000466' 262 17 0 00 000004                       RGET    (4,3,2,1,0)     ; restore AC's
000473' 263 17 0 00 000000                       RTN                     ; return
```

```

                                   ;#*************************************************************************
                                   ;* LSTAMP - Check for an 'S' typed.
                                   ;#*************************************************************************

000474' 261 17 0 00 000000         LSTAMP: RPUT    (0,1,2,3,4)                     ; save AC's
000501' 200 00 0 00 006126'                MOVE    LCHAR#                          ; get last character typed
000502' 306 00 0 00 000123                 CAIN    123                             ; was it an upper case-s?
000503' 254 00 0 00 000510'                JRST    LSTAMO                          ; yes - print current statistics
000504' 306 00 0 00 000163                 CAIN    163                             ; lower case-s?
000505' 254 00 0 00 000510'                JRST    LSTAMO                          ; yes - print current statistics
000506' 302 00 0 00 000023                 CAIE    023                             ; control-s?
000507' 254 00 0 00 000525'                JRST    LSTAMX                          ; no - exit

                                   ; Print runtime data

000510' 037 01 0 00 004273'        LSTAMO: FMSGC   <** Looping on Test >
000511' 200 01 0 00 000442*                MOVE    1,TSTCLS                        ; get test class
000512' 200 00 0 01 000443*                MOVE    INDLIS(1)                       ; get class description
000513' 037 01 0 00 000002                 PNTSXF                                  ; print it
000514' 200 00 0 00 000445*                MOVE    TSTNUM                          ; get test number
000515' 037 17 0 00 000003                 PNTOCF                                  ; print it
000516' 200 00 0 00 000447*                MOVE    TSTSUB                          ; get subtest number
000517' 322 00 0 00 000522'                JUMPE   .+3                             ; zero?  yes - continue
000520' 037 01 0 00 004073'                FMSG    <->                             ; no - print it
000521' 037 17 0 00 000003                 PNTOCF
000522' 037 01 0 00 004266'                FMSG    < at >
000523' 260 17 0 00 000317'                GO      PTIMEF                          ; print current time
000524' 037 01 0 00 004300'                FMSGD   < **>

000525' 402 00 0 00 006126'        LSTAMX: SETZM   LCHAR                           ; clear last character typed location
000526' 262 17 0 00 000004                 RGET    (4,3,2,1,0)                     ; restore AC's
000533' 263 17 0 00 000000                 RTN                                     ; return

```

```
                                                  SUBTTL  TOPS10/20 Related Routines

                                          ;#***************************************************************************
                                          ;* .RESET - Subroutine to issue a RESET JSYS/UUO
                                          ;
                                          ;   In exec mode it does nothing. In user mode it issues the appropriate
                                          ; RESET UUO/JSYS that may be desirable at the start or end of a program.
                                          ;#***************************************************************************

000534' 336 00 0 00 030037                .RESET: SKIPN   USER                    ; user mode?
000535' 263 17 0 00 000000                        RTN                             ; no - exit
000536' 336 00 0 00 030516                        SKIPN   MONTYP                  ; TOPS10?
000537' 047 00 0 00 000000                        CALLI   0                       ; do a RESET UUO
000540' 332 00 0 00 030516                        SKIPE   MONTYP                  ; TOPS20?
000541' 104 00 0 00 000147                        RESET                           ; do a RESET JSYS
000542' 263 17 0 00 000000                        RTN                             ; exit


                                          ;#***************************************************************************
                                          ;* .CLOSE - Subroutine to close opened files
                                          ;
                                          ;   This  does  nothing in exec mode.  In  user mode under TOPS10 and
                                          ; TOPS20 it takes the necessary steps to close  the output file which
                                          ; is necessary if  you've been outputting to a disk  file and wish to
                                          ; leave the program ....
                                          ;
                                          ;   Note *** Once you do a close from a Ctrl-C intercept the PNT File
                                          ; gets closed correctly but if you continue, the SUBRTN PKG creates a
                                          ; new PNT File and you lose the old one.
                                          ;#***************************************************************************

000543' 336 00 0 00 030037                .CLOSE: SKIPN   USER                    ; skip if we're in user mode
000544' 263 17 0 00 000000                        RTN                             ; exec mode
000545' 037 05 0 00 000002                        DROPDV                          ; SUBRTN PKG UUO or it won't work ok
000546' 261 17 0 00 000001                        PUT     1                       ; save AC
000547' 336 00 0 00 030516                        SKIPN   MONTYP                  ; TOPS20?
000550' 254 00 0 00 000555'                       JRST    CLS10                   ; no - go handle TOPS10
000551' 201 01 0 00 400000                        MOVEI   1,.FHSLF                ; get fork handle
000552' 104 00 0 00 000034                        CLZFF                           ; close all files JSYS ..
000553' 320 17 0 00 004247'                       ERCAL   [RTN]                   ; error return
000554' 254 00 0 00 000557'                       JRST    CLSX                    ; exit
000555' 070000  000000                    CLS10:  070000,,000000                  ; CLOSE D,0 UUO ... closes files
000556' 255 00 0 00 000000                        JFCL
000557' 262 17 0 00 000001                CLSX:   GET     1                       ; restore AC
000560' 263 17 0 00 000000                        RTN                             ;  and exit
```

```

                                        ;#**********************************************************************
                                        ;* .EXIT - Subroutine to exit job in user mode
                                        ;
                                        ; In exec mode it does nothing.  In user mode it issues the appropriate
                                        ; EXIT UUO or HALTF JSYS getting you back to monitor level.   This also
                                        ; allows you to continue with a continue command.
                                        ;
                                        ; Call  seq:
                                        ;       GO      .EXIT               ; the call
                                        ;       RTN1                        ; rtn+1 always
                                        ;#**********************************************************************

000561' 336 00 0 00 030037              .EXIT:  SKIPN   USER                ; user mode?
000562' 263 17 0 00 000000                      RTN                         ; no - exec mode - exit

000563' 261 17 0 00 000001                      PUT     1                   ; save AC1
000564' 336 00 0 00 030516                      SKIPN   MONTYP              ; TOPS20?
000565' 254 00 0 00 000572'                     JRST    EXT10               ; no - TOPS10

000566' 201 01 0 00 400000                      MOVEI   1,.FHSLF            ; get fork handle
000567' 104 00 0 00 000170                      HALTF                       ; halt... to monitor level
000570' 255 00 0 00 000000                      JFCL                        ; here on a continue command
000571' 254 00 0 00 000574'                     JRST    EXTX                ; to exit code

000572' 047040  000012                  EXT10:  047040,,000012              ; EXIT 1,  UUO  (CALLI  12 F=1)
000573' 255 00 0 00 000000                      JFCL                        ; here on a continue command

000574' 262 17 0 00 000001              EXTX:   GET     1                   ; restore the AC
000575' 263 17 0 00 000000                      RTN                         ; exit

```

```
                                                SUBTTL  File Handling Routines

                                ;#*********************************************************************
                                ;* FINIT - Initialize reading of microcode file
                                ;
                                ; Returns:      +1 - Error occurred
                                ;               +2 - OK
                                ;#*********************************************************************

000576' 261 17 0 00 000000      FINIT:  PUT     0
000577' 201 00 0 00 000657'             MOVEI   UNAME                   ; point to file name buffer
000600' 037 05 0 00 000004              FSELECT                         ; init the file
                                        JRST    [MOVEI  1,UNAME         ; error - get pointer to file name
                                                 GO     FFAIL           ; report to opr
                                                 GET    0               ; restore AC0
000601' 254 00 0 00 004302'                      RTN]                   ; exit
000602' 037 00 0 00 004306'             TMSGC   <[Reading file: >
000603' 200 00 0 00 000657'             MOVE    UNAME                   ; get the selected file name
000604' 037 00 0 00 000002              PNTSIX                          ; print it
000605' 037 00 0 00 000056              PNTCI   "."
000606' 200 00 0 00 000660'             MOVE    UNAME+1                 ; get the extension of file
000607' 037 00 0 00 000002              PNTSIX                          ; print it
000610' 037 00 0 00 004312'             TMSGD   <]>
000611' 262 17 0 00 000000              GET     0                       ; restore AC0
000612' 350 00 0 17 000000              AOS     (P)                     ; set up RTN+2
000613' 263 17 0 00 000000              RTN                             ; return


                                ;#*********************************************************************
                                ;* CHRIN - Routine to read an ASCII character from the selected file.
                                ;#*********************************************************************

000614' 037 06 0 00 000004      CHRIN:  FREAD                           ; tell DIAMON to get a file character
000615' 263 17 0 00 000000              RTN                             ; error or eof detected during reading
000616' 322 00 0 00 000614'             JUMPE   CHRIN                   ; eat all null characters
000617' 350 00 0 17 000000              AOS     (P)                     ; normal exit +2
000620' 263 17 0 00 000000              RTN
```

```
                                           ;#**********************************************************************
                                           ;* CHRINS - Routine to read an ASCII character from the selected file.
                                           ;#**********************************************************************

 000621' 261 17 0 00 000000        CHRINS: PUT     0                             ; save AC0
 000622' 037 06 0 00 000004                FREAD                                 ; get a file character
 000623' 254 00 0 00 000633'               JRST    CHRINX                        ; error or eof detected during reading
 000624' 322 00 0 00 000622'               JUMPE   .-2                           ; eat all null characters
 000625' 350 00 0 17 777777                AOS     -1(P)                         ; normal exit +2
 000626' 202 00 0 17 000000                MOVEM   (P)                           ; put character on stack
 000627' 332 00 0 00 000000*               SKIPE   FINECH                        ; echo flag set?
 000630' 037 12 0 00 000001                PNTCHF                                ; yes - echo the character
 000631' 262 17 0 00 000000                GET     0                             ; restore AC0
 000632' 263 17 0 00 000000                RTN

 000633' 037 01 0 00 004313'       CHRINX: FMSGC   <[End of >                    ; end of file
 000634' 200 00 0 00 000000*               MOVE    TAKFIL                        ; print file name
 000635' 037 01 0 00 000002                PNTSXF
 000636' 037 01 0 00 000056                PNTCIF  "."
 000637' 200 00 0 00 000000#               MOVE    TAKFIL+1
 000640' 037 01 0 00 000002                PNTSXF
 000641' 037 01 0 00 004312'               FMSGD   <]>
 000642' 402 00 0 00 000000*               SETZM   FINPUT                        ; clear 'file input' flag
 000643' 262 17 0 00 000000                GET     0                             ; restore AC0
 000644' 263 17 0 00 000000                RTN


                                           ;#**********************************************************************
                                           ;* FFAIL - Routine to report "DIAMON" open failure on a selected file.
                                           ;#**********************************************************************

 000645' 261 17 0 00 000000        FFAIL:  PUT     0                             ; save AC0
 000646' 037 01 0 00 004316'               FMSGC   <? Can't access ">
 000647' 200 00 0 00 000657'               MOVE    UNAME                         ; get file name
 000650' 037 01 0 00 000002                PNTSXF                                ; print it
 000651' 037 01 0 00 000056                PNTCIF  "."
 000652' 200 00 0 00 000660'               MOVE    UNAME+1                       ; get extension
 000653' 037 01 0 00 000002                PNTSXF                                ; print it
 000654' 037 01 0 00 004322'               FMSGD   <" on selected load device.>
 000655' 262 17 0 00 000000                GET     0                             ; restore AC0
 000656' 263 17 0 00 000000                RTN                                   ; return

 000657' 43 51 22 20 00 00         UNAME:  SIXBIT  /CI20/
 000660' 65 54 44 00 00 00                 SIXBIT  /ULD/
 749
 750                                              ;#*********************************************************************
 751                                              ;* FGETW - Get microword entry from file
 752                                              ;
 753                                              ; Arguments:  None (but file is already open and ready for use)
 754                                              ;
 755                                              ; Function:   Read a file entry consisting of '[ADDR]=Microword Data'
 756                                              ;
 757                                              ; Returns:    +1 - Error/EOF occurred
 758                                              ;             +2 - OK - Address is in CADDR
 759                                              ;                        Microword is in CWORDL,CWORDR
 760                                              ;
 761                                              ;#*********************************************************************
 762  000661' 261 17 0 00 000000          FGETW:  RPUT    (0,1,2)                  ; save AC's
 763  000664' 402 00 0 00 000000*                 SETZM   CADDR                    ; clear address
 764  000665' 402 00 0 00 000000*                 SETZM   CWORDL                   ; clear left half
 765  000666' 402 00 0 00 000000*                 SETZM   CWORDR                   ; clear right half
 766  000667' 260 17 0 00 000726'                 GO      FGETA                    ; get load address
 767  000670' 254 00 0 00 000722'                  JRST   FGETWE                   ; error/EOF occurred
 768  000671' 202 01 0 00 000664*                 MOVEM   1,CADDR                  ; save address
 769  000672' 260 17 0 00 000614'                 GO      CHRIN                    ; search for the = sign
 770  000673' 254 00 0 00 000722'                  JRST   FGETWE                   ; error/EOF occurred
 771  000674' 302 00 0 00 000075                  CAIE    "="                      ; equal sign?
 772  000675' 254 00 0 00 000672'                 JRST    .-3                      ; keep looking
 773  000676' 260 17 0 00 000750'                 GO      FGETD                    ; get a 4 digit number
 774  000677' 254 00 0 00 000722'                  JRST   FGETWE                   ; error/EOF occurred
 775  000700' 137 01 0 00 004330'                 DPB     1,[POINT 12,CWORDL,17]   ; save it
 776  000701' 260 17 0 00 000750'                 GO      FGETD                    ; get a 4 digit number
 777  000702' 254 00 0 00 000722'                  JRST   FGETWE                   ; error/EOF occurred
 778  000703' 137 01 0 00 004331'                 DPB     1,[POINT 12,CWORDL,29]   ; save it
 779  000704' 260 17 0 00 000750'                 GO      FGETD                    ; get a 4 digit number
 780  000705' 254 00 0 00 000722'                  JRST   FGETWE                   ; error/EOF occurred
 781  000706' 246 01 0 00 777772                  LSHC    1,-6                     ; right shift 6 bits
 782  000707' 137 01 0 00 004332'                 DPB     1,[POINT 6,CWORDL,35]    ; save 6 of the bits
 783  000710' 400 01 0 00 000000                  SETZ    1,                       ; clear AC1
 784  000711' 246 01 0 00 000006                  LSHC    1,6                      ; left shift 6 bits
 785  000712' 137 01 0 00 004333'                 DPB     1,[POINT 6,CWORDR,11]    ; save remaining 6 bits
 786  000713' 260 17 0 00 000750'                 GO      FGETD                    ; get a 4 digit number
 787  000714' 254 00 0 00 000722'                  JRST   FGETWE                   ; error/EOF occurred
 788  000715' 137 01 0 00 004334'                 DPB     1,[POINT 12,CWORDR,23]   ; save it
 789  000716' 260 17 0 00 000750'                 GO      FGETD                    ; get a 4 digit number
 790  000717' 254 00 0 00 000722'                  JRST   FGETWE                   ; error/EOF occurred
 791  000720' 137 01 0 00 004335'                 DPB     1,[POINT 12,CWORDR,35]   ; save it
 792  000721' 350 00 0 17 777775                  AOS     -3(P)                    ; done - set up RTN+2
 793  000722' 262 17 0 00 000002          FGETWE: RGET    (2,1,0)                  ; restore AC's
 794  000725' 263 17 0 00 000000                  RTN                              ; return
```

2086

```

                                        ;#*********************************************************************
                                        ;* FGETA - Obtain microcode load address 'ADDR:' from file
                                        ;
                                        ; Returns:    +1 - Error/EOF occurred
                                        ;             +2 - OK - AC1 contains octal address
                                        ;#*********************************************************************

000726' 261 17 0 00 000000              FGETA:  PUT     0                               ; save AC0
000727' 260 17 0 00 000614'             FGETA0: GO      CHRIN                           ; get a character
000730' 254 00 0 00 000740'                       JRST  FGETAE                          ; exit - error or EOF occurred
000731' 306 00 0 00 000073                      CAIN    ";"                             ; comment character?
000732' 260 17 0 00 000742'                     GO      FSKIPC                          ; yes - go skip it
000733' 302 00 0 00 000133                      CAIE    "["                             ; found beginning of address?
000734' 254 00 0 00 000727'                     JRST    FGETA0                          ; no - keep looking
000735' 260 17 0 00 000750'                     GO      FGETD                           ; get up to 4 digit # ended by CR or ]
000736' 254 00 0 00 000740'                       JRST  FGETAE                          ; exit - error or EOF occurred
000737' 350 00 0 17 777777                      AOS     -1(P)
000740' 262 17 0 00 000000              FGETAE: GET     0                               ; restore AC0
000741' 263 17 0 00 000000                      RTN                                     ; return

000742' 260 17 0 00 000614'             FSKIPC: GO      CHRIN                           ; get a character
000743' 254 00 0 00 000740'                       JRST  FGETAE                          ; exit - error or EOF occurred
000744' 302 00 0 00 000015                      CAIE    15                              ; CRLF yet?
000745' 306 00 0 00 000012                      CAIN    12
000746' 263 17 0 00 000000                      RTN                                     ; yes - return
000747' 254 00 0 00 000742'                     JRST    FSKIPC                          ; no - keep looking for it
```

2087

```
                                      ;#*********************************************************************
                                      ;* FGETD - Obtain microcode data entry from file (4 digits).
                                      ;
                                      ; Returns:    +1 - Error/EOF occurred
                                      ;             +2 - OK - AC1 contains octal data
                                      ;#*********************************************************************

000750' 261 17 0 00 000000    FGETD:  RPUT    (0,2)                  ; save AC's
000752' 400 01 0 00 000000            SETZ    1,                     ; clear result
000753' 201 02 0 00 000004            MOVEI   2,4                    ; up to 4 digits
000754' 260 17 0 00 000614'   FGETD0: GO      CHRIN                  ; get a digit
000755' 254 00 0 00 000772'            JRST   FGETDE                 ; error - exit
000756' 306 00 0 00 000015            CAIN    15                     ; carriage return?
000757' 254 00 0 00 000771'           JRST    FGETD1                 ; yes - done - exit
000760' 306 00 0 00 000135            CAIN    "]"                    ; end of address field?
000761' 254 00 0 00 000771'           JRST    FGETD1                 ; yes - done - exit
000762' 301 00 0 00 000060            CAIL    60                     ; is character between
000763' 303 00 0 00 000067            CAILE   67                     ;   0 and 7?
000764' 254 00 0 00 000772'           JRST    FGETDE                 ; no - error - exit
000765' 275 00 0 00 000060            SUBI    60                     ; normalize to 0-7
000766' 242 01 0 00 000003            LSH     1,3                    ; multiply result so far by 8
000767' 272 00 0 00 000001            ADDM    1                      ; add current digit
000770' 367 02 0 00 000754'           SOJG    2,FGETD0               ; loop till done
000771' 350 00 0 17 777776    FGETD1: AOS     -2(P)                  ; ok - set up RTN+2
000772' 262 17 0 00 000002    FGETDE: RGET    (2,0)                  ; restore AC's
000774' 263 17 0 00 000000            RTN                            ; return
```

```
                                                SUBTTL  Data Pattern Generator/Compare Routines

                                        ;#*****************************************************************
                                        ;* BUFGEN - Generate the specified data pattern in the data buffer.
                                        ;* BUFCOM - Compare the data buffer with expected data
                                        ;
                                        ; Arguments:  AC1 - Type:  1 read          -1 read (don't report errors)
                                        ;                          2 write         -2 write (don't report errors)
                                        ;                          3 read/write    -3 read/write (don't report errors)
                                        ;             AC2 - Length in words
                                        ;             PAT - Data pattern
                                        ;
                                        ; Returns (BUFCOM):  AC1 - Number of words in error
                                        ;#*****************************************************************

000775' 402 00 0 00 006136'             BUFCOM: SETZM   WRDERR                  ; clear words in error count
000776' 476 00 0 00 006122'                     SETOM   GENFLG#                 ; set to 'compare data'
000777' 476 00 0 00 006123'                     SETOM   GENREP#                 ; set 'report errors' flag
001000' 335 00 0 00 000001                      SKIPGE  1                       ; report errors?
001001' 402 00 0 00 006123'                     SETZM   GENREP                  ; no - clear 'report errors' flag
001002' 217 00 0 00 000001                      MOVMS   1                       ; get absolute value
001003' 306 01 0 00 000003                      CAIN    1,3                     ; read/write transfer?
001004' 254 00 0 00 001104'                     JRST    BUFCRW                  ; yes - go do the special compare
001005' 334 00 0 00 000000                      SKIPA
001006' 402 00 0 00 006122'             BUFGEN: SETZM   GENFLG                  ; set to 'generate data'
001007' 202 01 0 00 006124'                     MOVEM   1,GENTYP#               ; save type of transfer
001010' 322 14 0 00 004247'                     JUMPE   PAT,[RTN]               ; exit immediately if pat is zero
001011' 261 17 0 00 000000                      RPUT    (0,1,2,3,7,PAT)         ; save some AC's

                                        ; Find out fill pattern to use

001017' 201 00 0 00 000115                      MOVEI   FILLFP                  ; get fill pattern (fixed)
001020' 301 14 0 00 000116                      CAIL    PAT,VPATRN              ; variable pattern?
001021' 201 00 0 00 000127                      MOVEI   FILLVP                  ; yes - get fill pattern (variable)
001022' 202 00 0 00 006130'                     MOVEM   PATFLG#                 ; save it

                                        ; Check if read - if so set up fill pattern

001023' 332 00 0 00 006122'                     SKIPE   GENFLG                  ; compare data?
001024' 254 00 0 00 001027'                     JRST    .+3                     ; yes - continue
001025' 306 01 0 00 000001                      CAIN    1,1                     ; no - read?
001026' 200 14 0 00 006130'                     MOVE    PAT,PATFLG              ; yes - get fill pattern

                                        ; Handle buffer length - cannot exceed 512 words (all the CCWs necessary
                                        ; are generated by GENCCW, but they all start at the same address.

001027' 200 01 0 00 000002                      MOVE    1,2                     ; get number of full words in buffer
001030' 303 01 0 00 001000                      CAILE   1,^D512                 ; over 512 words?
001031' 201 01 0 00 001000                      MOVEI   1,^D512                 ; yes - limit to 512 words
001032' 201 02 0 00 000000*                     MOVEI   2,BUFF                  ; get buffer address
001033' 213 00 0 00 000001                      MOVNS   1                       ; generate AOBJN word to sequence
001034' 506 01 0 00 000002                      HRLM    1,2                     ;   through the buffer
```

```

                              ; Now generate/compare the data (for a constant pattern)
                              ; -------------------------------------------------------

001035' 301 14 0 00 000116            CAIL    PAT,VPATRN              ; skip if a constant pattern
001036' 254 00 0 00 001046'           JRST    BUFG1                   ; go generate variable pattern
001037' 200 07 0 14 001172'   BUFG0:  MOVE    7,PATBUF(PAT)           ; get pattern
001040' 336 00 0 00 006122'           SKIPN   GENFLG                  ; compare data?
001041' 202 07 0 02 000000            MOVEM   7,(2)                   ; no - put into buffer
001042' 332 00 0 00 006122'           SKIPE   GENFLG                  ; compare data?
                                      GO      [CAME   7,(2)           ; yes - compare it
                                               GO     DATMES          ; error - go handle
001043' 260 17 0 00 004336'                    RTN]

                              ; Now do next word

001044' 253 02 0 00 001037'           AOBJN   2,BUFG0                 ; loop till done
001045' 254 00 0 00 001061'           JRST    BUFGX                   ; all done - exit

                              ; Now generate/compare the data (for a variable pattern)
                              ; -------------------------------------------------------

                              ; Generate first word of buffer

001046' 200 07 0 14 001172'   BUFG1:  MOVE    7,PATBUF(PAT)           ; get pattern pointer word
001047' 552 07 0 00 006135'           HRRZM   7,VARPAT#               ; put adr of 2nd instruction in VARPAT
001050' 207 00 0 00 000007            MOVSS   7                       ; swap address of pointer
001051' 256 00 0 07 000000            XCT     (7)                     ; get first word of pattern
001052' 334 00 0 00 000000            SKIPA                           ; continue

                              ; Generate successive words of variable pattern

001053' 256 00 1 00 006135'   BUFG2:  XCT     @VARPAT                 ; generate another word of pattern
001054' 336 00 0 00 006122'           SKIPN   GENFLG                  ; compare data?
001055' 202 07 0 02 000000            MOVEM   7,(2)                   ; no - put into buffer
001056' 332 00 0 00 006122'           SKIPE   GENFLG                  ; compare data?
                                      GO      [CAME   7,(2)           ; yes - compare it
                                               GO     DATMES          ; error - go handle
001057' 260 17 0 00 004336'                    RTN]

                              ; Now do next word

001060' 253 02 0 00 001053'           AOBJN   2,BUFG2                 ; loop till done
```

```

                                ; All done - exit

001061' 336 00 0 00 006122'     BUFGX:  SKIPN   GENFLG                  ; compare data?
001062' 254 00 0 00 001075'             JRST    BUFGXZ                  ; no - exit
001063' 337 00 0 00 006136'             SKIPG   WRDERR#                 ; any compare errors?
001064' 254 00 0 00 001073'             JRST    BUFGXY                  ; no - return
001065' 336 00 0 00 006123'             SKIPN   GENREP                  ; report errors?
001066' 254 00 0 00 001073'             JRST    BUFGXY                  ; no - return
001067' 037 00 0 00 004341'             TMSGC   <Words in error = >
001070' 200 00 0 00 006136'             MOVE    WRDERR
001071' 037 15 0 00 000000              PNTDEC
001072' 037 00 0 00 004077'             TMSG    <.>
001073' 200 01 0 00 006136'     BUFGXY: MOVE    1,WRDERR                ; get number of errors
001074' 202 01 0 17 777774              MOVEM   1,-4(P)                 ; save it
001075' 262 17 0 00 000014      BUFGXZ: RGET    (PAT,7,3,2,1,0)         ; restore AC's
001103' 263 17 0 00 000000              RTN

                                ; Data compare for split buffer

001104' 261 17 0 00 000000      BUFCRW: RPUT    (0,1,2,3,7,PAT,10)      ; save some AC's
001113' 200 01 0 00 000002              MOVE    1,2                     ; get number of full words in buffer
001114' 242 01 0 00 777777              LSH     1,-1                    ; divide by 2
001115' 201 10 0 00 001032*             MOVEI   10,BUFF                 ; get buffer address
001116' 270 10 0 00 000001              ADD     10,1                    ; point to middle of buffer
001117' 201 02 0 00 001115*             MOVEI   2,BUFF                  ; get buffer address
001120' 213 00 0 00 000001              MOVNS   1                       ; generate AOBJN word to sequence
001121' 506 01 0 00 000002              HRLM    1,2                     ;   through the buffer

                                ; Do the data compare

001122' 550 07 0 00 000002      BUFCR1: HRRZ    7,2                     ; get address portion
001123' 200 07 0 10 000000              MOVE    7,(10)                  ; get word
001124' 312 07 0 02 000000              CAME    7,(2)                   ; data compares?
001125' 260 17 0 00 001132'             GO      DATMES                  ; no - go handle error

                                ; Now do next word

001126' 350 00 0 00 000010              AOS     10                      ; point to next word
001127' 253 02 0 00 001122'             AOBJN   2,BUFCR1                ; loop till done
001130' 262 17 0 00 000010              GET     10                      ; restore AC10
001131' 254 00 0 00 001061'             JRST    BUFGX                   ; all done - exit
```

```
                                 ;#********************************************************************
                                 ;* DATMES - A data comparison error has been found.  Report the error,
                                 ;* printing a maximum of three lines of errors.
                                 ;#********************************************************************

001132' 261 17 0 00 000000       DATMES: RPUT    (0,1,2,3)               ; save AC's
001136' 336 00 0 00 006123'              SKIPN   GENREP                  ; report errors?
001137' 254 00 0 00 001140'              JRST    DATMO                   ; no - skip this
001140' 350 03 0 00 006136'      DATMO:  AOS     3,WRDERR                ; increment error count

                                 ; Now see if should be printed

001141' 336 00 0 00 006123'              SKIPN   GENREP                  ; report errors?
001142' 254 00 0 00 001165'              JRST    DATMX                   ; no - exit
001143' 307 03 0 00 000003               CAIG    3,3                     ; already print 3 of them?
001144' 254 00 0 00 001146'              JRST    DATM1                   ; no - go print
001145' 254 00 0 00 001165'              JRST    DATMX                   ; yes - exit

                                 ; Now print

001146' 302 03 0 00 000001       DATM1:  CAIE    3,1                     ; first data error?
001147' 254 00 0 00 001152'              JRST    .+3                     ; no - don't print headers ...
001150' 037 00 0 00 004345'              TMSGC   <Data Compare Error:>
001151' 037 00 0 00 004352'              TMSGC   <Word    ---Correct---    ---Actual---->
001152' 550 00 0 00 000002               HRRZ    2                       ; get word number, numbered
001153' 275 00 0 00 000000#              SUBI    BUFF-1                  ;   from 1 ...
001154' 202 00 0 00 006115'              MOVEM   CWORDN#                 ; save current word number
001155' 037 00 0 00 030242               PCRL
001156' 006 04 0 00 000000               PDEC    4,0                     ; print it
001157' 037 00 0 00 004107'              TMSG    <   >                   ; print spaces
001160' 200 00 0 00 000007               MOVE    7                       ; get correct data
001161' 037 13 0 00 000000               PNTHW                           ; print it
001162' 037 00 0 00 004106'              TMSG    < >                     ; print some more spaces
001163' 200 00 0 02 000000               MOVE    (2)                     ; get actual data
001164' 037 13 0 00 000000               PNTHW                           ; print it

                                 ; Exit

001165' 262 17 0 00 000003       DATMX:  RGET    (3,2,1,0)               ; restore AC's
001171' 263 17 0 00 000000               RTN                             ; return
```

```
                                                 SUBTTL  Data Patterns

                                         ;#*****************************************************************
                                         ;* Data Patterns
                                         ;
                                         ; Here are the locations called by the BUFGEN and BUFCOM routines
                                         ; indexed by the pattern number.
                                         ;#*****************************************************************

001172' 000000  000000                   PATBUF: 0                              ; unused
001173' 000000  000000                           0                              ; 1 - zeros
001174' 777777  777777                           -1                             ; 2 - ones
001175' 525252  525252                           525252525252                   ; 3 - alternate bit in 36-bit word
001176' 252525  252525                           252525252525                   ; 4 - alternate bit in 36-bit word
001177' 000000  000001                           1                              ; 5 - floating 1's
001200' 000000  000002                           2                              ; 6
001201' 000000  000004                           4                              ; 7
001202' 000000  000010                           10                             ; 10
001203' 000000  000020                           20                             ; 11
001204' 000000  000040                           40                             ; 12
001205' 000000  000100                           100                            ; 13
001206' 000000  000200                           200                            ; 14
001207' 000000  000400                           400                            ; 15
001210' 000000  001000                           1000                           ; 16
001211' 000000  002000                           2000                           ; 17
001212' 000000  004000                           4000                           ; 20
001213' 000000  010000                           10000                          ; 21
001214' 000000  020000                           20000                          ; 22
001215' 000000  040000                           40000                          ; 23
001216' 000000  100000                           100000                         ; 24
001217' 000000  200000                           200000                         ; 25
001220' 000000  400000                           400000                         ; 26
001221' 000001  000000                           1,,0                           ; 27
001222' 000002  000000                           2,,0                           ; 30
001223' 000004  000000                           4,,0                           ; 31
001224' 000010  000000                           10,,0                          ; 32
001225' 000020  000000                           20,,0                          ; 33
001226' 000040  000000                           40,,0                          ; 34
001227' 000100  000000                           100,,0                         ; 35
001230' 000200  000000                           200,,0                         ; 36
001231' 000400  000000                           400,,0                         ; 37
001232' 001000  000000                           1000,,0                        ; 40
001233' 002000  000000                           2000,,0                        ; 41
001234' 004000  000000                           4000,,0                        ; 42
001235' 010000  000000                           10000,,0                       ; 43
001236' 020000  000000                           20000,,0                       ; 44
001237' 040000  000000                           40000,,0                       ; 45
001240' 100000  000000                           100000,,0                      ; 46
001241' 200000  000000                           200000,,0                      ; 47
001242' 400000  000000                           400000,,0                      ; 50
```

```
 1079  001243' 777777  777776                 777777,,777776          ; 51 - floating 0's
 1080  001244' 777777  777775                 777777,,777775          ; 52
 1081  001245' 777777  777773                 777777,,777773          ; 53
 1082  001246' 777777  777767                 777777,,777767          ; 54
 1083  001247' 777777  777757                 777777,,777757          ; 55
 1084  001250' 777777  777737                 777777,,777737          ; 56
 1085  001251' 777777  777677                 777777,,777677          ; 57
 1086  001252' 777777  777577                 777777,,777577          ; 60
 1087  001253' 777777  777377                 777777,,777377          ; 61
 1088  001254' 777777  776777                 777777,,776777          ; 62
 1089  001255' 777777  775777                 777777,,775777          ; 63
 1090  001256' 777777  773777                 777777,,773777          ; 64
 1091  001257' 777777  767777                 777777,,767777          ; 65
 1092  001260' 777777  757777                 777777,,757777          ; 66
 1093  001261' 777777  737777                 777777,,737777          ; 67
 1094  001262' 777777  677777                 777777,,677777          ; 70
 1095  001263' 777777  577777                 777777,,577777          ; 71
 1096  001264' 777777  377777                 777777,,377777          ; 72
 1097  001265' 777776  777777                 777776,,777777          ; 73
 1098  001266' 777775  777777                 777775,,777777          ; 74
 1099  001267' 777773  777777                 777773,,777777          ; 75
 1100  001270' 777767  777777                 777767,,777777          ; 76
 1101  001271' 777757  777777                 777757,,777777          ; 77
 1102  001272' 777737  777777                 777737,,777777          ; 100
 1103  001273' 777677  777777                 777677,,777777          ; 101
 1104  001274' 777577  777777                 777577,,777777          ; 102
 1105  001275' 777377  777777                 777377,,777777          ; 103
 1106  001276' 776777  777777                 776777,,777777          ; 104
 1107  001277' 775777  777777                 775777,,777777          ; 105
 1108  001300' 773777  777777                 773777,,777777          ; 106
 1109  001301' 767777  777777                 767777,,777777          ; 107
 1110  001302' 757777  777777                 757777,,777777          ; 110
 1111  001303' 737777  777777                 737777,,777777          ; 111
 1112  001304' 677777  777777                 677777,,777777          ; 112
 1113  001305' 577777  777777                 577777,,777777          ; 113
 1114  001306' 377777  777777                 377777,,777777          ; 114
 1115
 1116                  000115         FILLFP==.-PATBUF                 ; read fill pattern
 1117
 1118  001307' 400000  000001                 400000,,000001          ; 115
 1119
 1120                                 ; Pattern codes from here require execution of special routines
 1121
 1122                  000116         VPATRN==.-PATBUF
 1123
 1124                                 ; Each location contains addresses, the first address should be used
 1125                                 ; to generate the first word in a buffer, the second  should be used
 1126                                 ; to generate each remaining word.  An execute is  performed  on the
 1127                                 ; instruction in the specified address.  The data is returned in AC7.
 1128
 1129  001310' 004362' 004363'        [GO PATRAN],,[GO PATRA1]        ; 116 - random numbers
 1130  001311' 004364' 004366'        [GO PATADR],,[ADD 7,[1,,1]]     ; 117 - memory address pattern
 1131  001312' 004367' 004370'        [MOVEI 7,1],,[ROT 7,1]          ; 120 - floating ones
 1132  001313' 004371' 004370'        [SETCMI 7,1],,[ROT 7,1]         ; 121 - floating zeros
```

```
DFPTAU  MA
DFPTAU  MA

 1646
 1647
 1648
 1649
 1650
 1651  002
 1652  002
 1653  002
 1654  002
 1655  002
 1656  002
 1657  002
 1658  002
 1659  002
 1660  002
 1661  002
 1662  002
 1663  002
 1664  002
 1665
 1666  002
 1667  002
 1668  002
 1669  002
 1670  002
 1671  002
 1672  002
 1673  002
 1674  002
 1675  002
 1676  002
 1677  00
 1678  002
 1679  002
```

```

                           004372'        PATCN2==[BYTE (8)1,2,3,4]
                           004373'        PATCN3==[BYTE (7)1,2,3,4,5]
                           004374'        PATCN4==[BYTE (6)1,2,3,4,5,6]

        001314' 004375' 004376'                   [MOVE 7,PATCN2],,[GO PATCN8]   ; 122 - count pattern in 8-bit bytes
        001315' 004377' 004400'                   [MOVE 7,PATCN3],,[GO PATCN7]   ; 123 - count pattern in 7-bit bytes
        001316' 004401' 004402'                   [MOVE 7,PATCN4],,[GO PATCN6]   ; 124 - count pattern in 6-bit bytes
        001317' 004403' 004404'                   [MOVE 7,PATBUF+3],,[SETCA 7,]  ; 125 - alternating 36-bit pattern
        001320' 004405' 004131'                   [MOVE 7,TPAT],,[JFCL]          ; 126 - operator selected pattern

                           000127         FILLVP==.-PATBUF                       ; read fill pattern

        001321' 004407' 004131'                   [MOVE 7,[400000,,1]],,[JFCL]   ; 127 - read buffer fill pattern

                           000127         MAXPAT==.-PATBUF-1                     ; maximum pattern number allowed

                                          ; Random number generator

        001322' 200 07 0 00 001323'       PATRAN: MOVE    7,.+1                  ; get base number
        001323' 202 07 0 00 006127'               MOVEM   7,OLDRAN#              ; save it
        001324' 263 17 0 00 000000                RTN

        001325' 270 07 0 00 030022        PATRA1: ADD     7,RANDBS               ; modify number in AC7
        001326' 241 07 0 00 777774                ROT     7,-4                   ; to create a new number
        001327' 447 07 0 00 006127'               EQVB    7,OLDRAN               ; put result in AC7 and OLDRAN
        001330' 263 17 0 00 000000                RTN

                                          ; Put address of data pattern in AC7

        001331' 201 07 0 00 001117*       PATADR: MOVEI   7,BUFF                 ; get buffer address
        001332' 504 07 0 00 000007                HRL     7,7                    ; put in both halves
        001333' 263 17 0 00 000000                RTN

                                          ; Count the number in AC7 in 8-bit bytes

        001334' 261 17 0 00 000005        PATCN8: RPUT    (5,6)                  ; save AC's
        001336' 201 05 0 00 000004                MOVEI   5,4                    ; set up to loop 4 times
        001337' 135 06 0 00 004410'               LDB     6,[POINT 8,7,31]       ; get last byte into AC6
        001340' 350 00 0 00 000006                AOS     6                      ; increment byte number
        001341' 405 06 0 00 000377                ANDI    6,377                  ; clear all but one byte
        001342' 242 07 0 00 000010                LSH     7,8                    ; shift word
        001343' 436 06 0 00 000007                ORM     6,7                    ; merge with new byte
        001344' 367 05 0 00 001340'               SOJG    5,.-4                  ; loop on 5 bytes
        001345' 242 07 0 00 000004                LSH     7,4                    ; shift over and clear bit 32-35
        001346' 262 17 0 00 000006                RGET    (6,5)                  ; restore AC's
        001350' 263 17 0 00 000000                RTN                            ; return
```

```
                                                ; Count the number in AC7 in 7-bit bytes

001351' 261 17 0 00 000005      PATCN7: RPUT    (5,6)                   ; save AC's
001353' 201 05 0 00 000005              MOVEI   5,5                     ; set up to loop 5 times
001354' 135 06 0 00 004411'             LDB     6,[POINT 7,7,34]        ; get last byte into AC6
001355' 350 00 0 00 000006              AOS     6                       ; increment byte number
001356' 405 06 0 00 000177              ANDI    6,177                   ; clear all but one byte
001357' 242 07 0 00 000007              LSH     7,7                     ; shift word
001360' 436 06 0 00 000007              ORM     6,7                     ; merge with new byte
001361' 367 05 0 00 001355'             SOJG    5,.-4                   ; loop on 5 bytes
001362' 242 07 0 00 000001              LSH     7,1                     ; shift over and clear bit 35
001363' 262 17 0 00 000006              RGET    (6,5)                   ; restore AC's
001365' 263 17 0 00 000000              RTN                             ; return

                                        ; Count the number in AC7 in 6-bit bytes

001366' 261 17 0 00 000005      PATCN6: RPUT    (5,6)                   ; save AC's
001370' 201 05 0 00 000006              MOVEI   5,6                     ; set up to loop 6 times
001371' 550 06 0 00 000007              HRRZ    6,7                     ; get word into AC6
001372' 350 00 0 00 000006              AOS     6                       ; increment byte number
001373' 405 06 0 00 000077              ANDI    6,77                    ; clear all but one byte
001374' 242 07 0 00 000006              LSH     7,6                     ; shift word
001375' 436 06 0 00 000007              ORM     6,7                     ; merge with new byte
001376' 367 05 0 00 001372'             SOJG    5,.-4                   ; loop on four bytes
001377' 262 17 0 00 000006              RGET    (6,5)                   ; restore AC's
001401' 263 17 0 00 000000              RTN                             ; return
```

```


                                    ;#*********************************************************************
                                    ;* PATPNT - Translate pattern number to description
                                    ;#*********************************************************************

001402' 037 00 0 00 004412'         PATPNT: TMSG    <0 - Illegal>           ; unused
001403' 037 00 0 00 004415'                 TMSG    <1 - zeros>
001404' 037 00 0 00 004417'                 TMSG    <2 - ones>
001405' 037 00 0 00 004421'                 TMSG    <3 - alternating 5252..>
001406' 037 00 0 00 004426'                 TMSG    <4 - alternating 2525..>
001407' 037 00 0 00 004433'                 TMSG    <5 - 0,,1>
001410' 037 00 0 00 004435'                 TMSG    <6 - 0,,2>
001411' 037 00 0 00 004437'                 TMSG    <7 - 0,,4>
001412' 037 00 0 00 004441'                 TMSG    <10 - 0,,10>
001413' 037 00 0 00 004444'                 TMSG    <11 - 0,,20>
001414' 037 00 0 00 004447'                 TMSG    <12 - 0,,40>
001415' 037 00 0 00 004452'                 TMSG    <13 - 0,,100>
001416' 037 00 0 00 004455'                 TMSG    <14 - 0,,200>
001417' 037 00 0 00 004460'                 TMSG    <15 - 0,,400>
001420' 037 00 0 00 004463'                 TMSG    <16 - 0,,1000>
001421' 037 00 0 00 004466'                 TMSG    <17 - 0,,2000>
001422' 037 00 0 00 004471'                 TMSG    <20 - 0,,4000>
001423' 037 00 0 00 004474'                 TMSG    <21 - 0,,10000>
001424' 037 00 0 00 004477'                 TMSG    <22 - 0,,20000>
001425' 037 00 0 00 004502'                 TMSG    <23 - 0,,40000>
001426' 037 00 0 00 004505'                 TMSG    <24 - 0,,100000>
001427' 037 00 0 00 004510'                 TMSG    <25 - 0,,200000>
001430' 037 00 0 00 004513'                 TMSG    <26 - 0,,400000>
001431' 037 00 0 00 004516'                 TMSG    <27 - 1,,0>
001432' 037 00 0 00 004520'                 TMSG    <30 - 2,,0>
001433' 037 00 0 00 004522'                 TMSG    <31 - 4,,0>
001434' 037 00 0 00 004524'                 TMSG    <32 - 10,,0>
001435' 037 00 0 00 004527'                 TMSG    <33 - 20,,0>
001436' 037 00 0 00 004532'                 TMSG    <34 - 40,,0>
001437' 037 00 0 00 004535'                 TMSG    <35 - 100,,0>
001440' 037 00 0 00 004540'                 TMSG    <36 - 200,,0>
001441' 037 00 0 00 004543'                 TMSG    <37 - 400,,0>
001442' 037 00 0 00 004546'                 TMSG    <40 - 1000,,0>
001443' 037 00 0 00 004551'                 TMSG    <41 - 2000,,0>
001444' 037 00 0 00 004554'                 TMSG    <42 - 4000,,0>
001445' 037 00 0 00 004557'                 TMSG    <43 - 10000,,0>
001446' 037 00 0 00 004562'                 TMSG    <44 - 20000,,0>
001447' 037 00 0 00 004565'                 TMSG    <45 - 40000,,0>
001450' 037 00 0 00 004570'                 TMSG    <46 - 100000,,0>
001451' 037 00 0 00 004573'                 TMSG    <47 - 200000,,0>
001452' 037 00 0 00 004576'                 TMSG    <50 - 400000,,0>
```

```
  1253  001453' 037 00 0 00 004601'           TMSG    <51 - 777777,,777776>
  1254  001454' 037 00 0 00 004605'           TMSG    <52 - 777777,,777775>
  1255  001455' 037 00 0 00 004611'           TMSG    <53 - 777777,,777773>
  1256  001456' 037 00 0 00 004615'           TMSG    <54 - 777777,,777767>
  1257  001457' 037 00 0 00 004621'           TMSG    <55 - 777777,,777757>
  1258  001460' 037 00 0 00 004625'           TMSG    <56 - 777777,,777737>
  1259  001461' 037 00 0 00 004631'           TMSG    <57 - 777777,,777677>
  1260  001462' 037 00 0 00 004635'           TMSG    <60 - 777777,,777577>
  1261  001463' 037 00 0 00 004641'           TMSG    <61 - 777777,,777377>
  1262  001464' 037 00 0 00 004645'           TMSG    <62 - 777777,,776777>
  1263  001465' 037 00 0 00 004651'           TMSG    <63 - 777777,,775777>
  1264  001466' 037 00 0 00 004655'           TMSG    <64 - 777777,,773777>
  1265  001467' 037 00 0 00 004661'           TMSG    <65 - 777777,,767777>
  1266  001470' 037 00 0 00 004665'           TMSG    <66 - 777777,,757777>
  1267  001471' 037 00 0 00 004671'           TMSG    <67 - 777777,,737777>
  1268  001472' 037 00 0 00 004675'           TMSG    <70 - 777777,,677777>
  1269  001473' 037 00 0 00 004701'           TMSG    <71 - 777777,,577777>
  1270  001474' 037 00 0 00 004705'           TMSG    <72 - 777777,,377777>
  1271  001475' 037 00 0 00 004711'           TMSG    <73 - 777776,,777777>
  1272  001476' 037 00 0 00 004715'           TMSG    <74 - 777775,,777777>
  1273  001477' 037 00 0 00 004721'           TMSG    <75 - 777773,,777777>
  1274  001500' 037 00 0 00 004725'           TMSG    <76 - 777767,,777777>
  1275  001501' 037 00 0 00 004731'           TMSG    <77 - 777757,,777777>
  1276  001502' 037 00 0 00 004735'           TMSG    <100 - 777737,,777777>
  1277  001503' 037 00 0 00 004742'           TMSG    <101 - 777677,,777777>
  1278  001504' 037 00 0 00 004747'           TMSG    <102 - 777577,,777777>
  1279  001505' 037 00 0 00 004754'           TMSG    <103 - 777377,,777777>
  1280  001506' 037 00 0 00 004761'           TMSG    <104 - 776777,,777777>
  1281  001507' 037 00 0 00 004766'           TMSG    <105 - 775777,,777777>
  1282  001510' 037 00 0 00 004773'           TMSG    <106 - 773777,,777777>
  1283  001511' 037 00 0 00 005000'           TMSG    <107 - 767777,,777777>
  1284  001512' 037 00 0 00 005005'           TMSG    <110 - 757777,,777777>
  1285  001513' 037 00 0 00 005012'           TMSG    <111 - 737777,,777777>
  1286  001514' 037 00 0 00 005017'           TMSG    <112 - 677777,,777777>
  1287  001515' 037 00 0 00 005024'           TMSG    <113 - 577777,,777777>
  1288  001516' 037 00 0 00 005031'           TMSG    <114 - 377777,,777777>
  1289  001517' 037 00 0 00 005036'           TMSG    <115 - 400000,,1>
  1290  001520' 037 00 0 00 005042'           TMSG    <116 - random numbers>
  1291  001521' 037 00 0 00 005047'           TMSG    <117 - memory address pattern>
  1292  001522' 037 00 0 00 005055'           TMSG    <120 - floating ones>
  1293  001523' 037 00 0 00 005061'           TMSG    <121 - floating zeros>
  1294  001524' 037 00 0 00 005066'           TMSG    <122 - count pattern in 8-bit bytes>
  1295  001525' 037 00 0 00 005075'           TMSG    <123 - count pattern in 7-bit bytes>
  1296  001526' 037 00 0 00 005104'           TMSG    <124 - count pattern in 6-bit bytes>
  1297  001527' 037 00 0 00 005113'           TMSG    <125 - alternating 36-bit pattern>
  1298  001530' 037 00 0 00 005122'           TMSG    <126 - opr select pattern>
  1299  001531' 037 00 0 00 005127'           TMSG    <127 - 400000,,1>
  1300
```

```
                                                 SUBTTL  Switch Handling Routines

                                         ;#*******************************************************************
                                         ;* .SWCHP - Print switch settings in English
                                         ;#*******************************************************************

001532' 261 17 0 00 000000               .SWCHP: RPUT    (0,1,2,3,4)             ; save AC's
001537' 037 01 0 00 005133'                      FMSGC   <Switches:  >
001540' 403 02 0 00 000004                       SETZB   2,4                     ; init print, switch count AC's
001541' 400 03 0 00 000000                       SETZ    3,                      ; set up AC3 to point to
001542' 661 03 0 00 400000                       TLO     3,(1B0)                 ;   the leftmost switch
001543' 260 17 0 00 001571'                      GO      SWITT                   ; get program switches
001544' 037 13 0 00 000001                       PNTHWF                          ; print in octal first
001545' 037 01 0 00 004106'                      FMSG    <  >
001546' 260 17 0 00 001571'                      GO      SWITT                   ; get program switches

001547' 616 00 0 00 000003               .SWCH0: TDNN    3                       ; this switch set?
001550' 254 00 0 00 001557'                      JRST    .SWCH1                  ; no - continue
001551' 271 04 0 00 000007                       ADDI    4,7                     ; 6 more characters printed
001552' 303 04 0 00 000052                       CAILE   4,^D42                  ; print 42 characters yet?
001553' 037 01 0 00 005136'                      FMSGC   <
                                                           > ; yes - continue on next line
001554' 303 04 0 00 000052                       CAILE   4,^D42                  ; print 42 characters yet?
001555' 201 04 0 00 000007                       MOVEI   4,7                     ; yes - reinit print counter
                                                 XCT     [FMSG   <ABORT >        ; 0
                                                          FMSG   <RSTART >       ; 1
                                                          FMSG   <TOTALS >       ; 2
                                                          FMSG   <NOPNT >        ; 3
                                                          FMSG   <PNTLPT >       ; 4
                                                          FMSG   <DING >         ; 5
                                                          FMSG   <LOOPER >       ; 6
                                                          FMSG   <ERSTOP >       ; 7
                                                          FMSG   <PALERS >       ; 8
                                                          FMSG   <RELIAB >       ; 9
                                                          FMSG   <TXTINH >       ; 10
                                                          FMSG   <PAGINH >       ; 11
                                                          FMSG   <MODDVC >       ; 12
                                                          FMSG   <CASINH >       ; 13
                                                          FMSG   <OPRSEL >       ; 14
                                                          FMSG   <CHAIN >        ; 15
                                                          FMSG   <KA10 >         ; 16
                                                          JFCL
                                                          FMSG   <TRACE >        ; 18
                                                          FMSG   <INHFLT >       ; 19
                                                          FMSG   <INHMSG >       ; 20
                                                          FMSG   <LOOPGM >       ; 21
                                                          FMSG   <LOOPTS >       ; 22
                                                          FMSG   <RUNALL >       ; 23
                                                          FMSG   <DSPEAR >       ; 24
```

```
                                                           FMSG   <UDEBUG >        ; 25
                                                   FMSG   <MDEBUG >        ; 26
                                                   FMSG   <SDEBUG >        ; 27
                                                   FMSG   <LDEBUG >        ; 28
                                                   FMSG   <IDEBUG >        ; 29
                                                   FMSG   <NIPORT >        ; 30
                                                   FMSG   <CIPORT >        ; 31
                                                   JFCL                    ; 32
                                                   JFCL                    ; 33
                                                   FMSG   <MMPROC >        ; 34
001556' 256 00 0 02 005242'                        FMSG   <MCBUS  >](2)    ; 35
001557' 242 03 0 00 777777      .SWCH1: LSH     3,-1                      ; right shift the switch pointer
001560' 350 00 0 00 000002              AOS     2                         ; increment text pointer
001561' 326 03 0 00 001547'             JUMPN   3,.SWCH0                  ; done?  no - continue
001562' 037 01 0 00 030242              PCRLF
001563' 262 17 0 00 000004              RGET    (4,3,2,1,0)               ; restore AC's
001570' 263 17 0 00 000000              RTN                               ; return



                                ;#*****************************************************************
                                ;* SWITT - Get value of switches
                                ;#*****************************************************************

001571' 200 00 0 00 030046      SWITT:  MOVE    CONSW                     ; get switches
001572' 336 00 0 00 030037              SKIPN   USER                      ; user mode?
001573' 037 10 0 00 000002              SWITCH                            ; no - get switches from KLDCP
001574' 434 00 0 00 001576'             IOR     SWRGT                     ; insert addl right hand switches
001575' 263 17 0 00 000000              RTN                               ; return

001576' 000000  000000          SWRGT:  0                                 ; right hand switches

```

```

                                ;#*********************************************************************
                                ;* .ISWT - Complete switch handling
                                ;
                                ; This routine will print out switches if desired, and then enter left
                                ; hand switches (if user mode) and then enter right hand switches.
                                ;
                                ;#*********************************************************************

                                ; Print program switches if desired

 001577' 261 17 0 00 000000     .ISWT:  RPUT    (0,1)                   ; save AC's
 001601' 260 17 0 00 001637'            GO      .SWPNT                  ; print switches if desired

                                ; Now look at switches and determine if we should enter them

 001602' 260 17 0 00 001571'    .ISWT2: GO      SWITT                   ; get switches
 001603' 202 00 0 00 006131'            MOVEM   SAVSWW#                 ; save switches
 001604' 202 00 0 00 006113'            MOVEM   CONSWW#                 ; save in work location
 001605' 037 01 0 00 005306'            FMSGC   <Switches = >
 001606' 037 13 0 00 000001             PNTHWF
 001607' 037 01 0 00 030242             PCRLF
 001610' 476 00 0 00 006134'            SETOM   SWPSWW#                 ; set switches printed flag
 001611' 336 00 0 00 030037             SKIPN   USER                    ; user mode?
 001612' 254 00 0 00 001614'            JRST    .ISWT7                  ; no - don't ask for left hand switches

                                ; Now input the switches

 001613' 260 17 0 00 002254'    .ISWT6: GO      .SLEFT                  ; get left hand switches

                                ; Now get right hand switches

 001614' 332 00 0 00 030037     .ISWT7: SKIPE   USER                    ; user mode?
 001615' 254 00 0 00 001620'            JRST    .ISWTB                  ; yes - go ask for them

                                ; In exec mode get switches which may have been entered already

 001616' 260 17 0 00 001571'    .ISWT8: GO      SWITT                   ; get switches
 001617' 202 00 0 00 006113'            MOVEM   CONSWW                  ; save switches

                                ; Now input the switches

 001620' 260 17 0 00 002355'    .ISWTB: GO      .SRIGT                  ; go get right hand switches

                                ; Now ensure right hand switches are saved and print out new switches
                                ; if they changed.

 001621' 200 00 0 00 006113'    .ISWTX: MOVE    CONSWW                  ; get switches
 001622' 552 00 0 00 001576'            HRRZM   SWRGT                   ; make sure right half is saved away
 001623' 202 00 0 00 030046             MOVEM   CONSW                   ; save in real switches location also
```

```
001624' 260 17 0 00 001571'              GO      SWITT                          ; get switches
001625' 312 00 0 00 006131'              CAME    SAVSWW                         ; same as original switches?
001626' 254 00 0 00 001631'              JRST    .ISWTY                         ; no - go print them out
001627' 332 00 0 00 006134'              SKIPE   SWPSWW#                        ; switches already printed?
001630' 254 00 0 00 001634'              JRST    .ISWTZ                         ; yes - don't print them
001631' 037 01 0 00 005311'      .ISWTY: FMSGC   <SWITCHES = >
001632' 037 13 0 00 000001               PNTHWF
001633' 037 01 0 00 030242               PCRLF
001634' 262 17 0 00 000001       .ISWTZ: RGET    (1,0)                          ; done with switch handling
001636' 263 17 0 00 000000               RTN                                    ; return

                                 ; Print switches available

001637' 037 01 0 00 000026       .SWPNT: PFORCE                                 ; handle Control-O
001640' 200 00 0 00 005324'              MOVE    [FMSGC  <Print the selectable pgm switches?>]
001641' 260 17 0 00 000061'              GO      TTYYES                         ; go get response
001642' 263 17 0 00 000000               RTN                                    ; no - return
001643' 037 00 0 00 001645'              PNTMSG  .SWTXT                         ; yes - print it
001644' 263 17 0 00 000000               RTN                                    ; return

001645'                          .SWTXT: ASCIZ   /
001645' 015 012 055 055 055      ----Left Side Switches--------        ----Right Side Switches--------

                                 SW    KL10    11     Function          SW  Value   Function
                                 --    ----    ----   --------          --  ------  --------
                                  0   400000  100000  Abort             18  400000  Program trace
                                  1   200000  040000  Restart           19  200000  Inhibit fault isolation
                                  2   100000  020000  Totals            20  100000  Inhibit error messages
                                  3   040000  010000  No print          21  040000  Loop on program
                                  4   020000  004000  Print on LPT      22  020000  Loop on test
                                  5   010000  002000  Ding bell on err  23  010000  Run all test segments
                                  6   004000  001000  Loop on error     24  004000  Disable SPEAR reporting
                                  7   002000  000400  Halt on error     25  002000  User mode debug
                                  8   001000  000200  Print all errors  26  001000  Error message debug
                                  9   000400  000100  Reliability mode  27  000400  Test debug
                                 10   000200  000040  Text inhibit      28  000200  Load microcode debug
                                 11   000100  000020  Inhibit paging    29  000100  Fault isolation debug
                                 12   000040  000010  Mod device codes  30  000040  NI port
                                 13   000020  000004  Inhibit cache     31  000020  CI port
                                 14   000010  000002  Operator select   34  000002  Missing MPROC module
                                 15   000004  000001  Chain.. (DIAMON)  35  000001  Missing CBUS module
                                 16   000002          KA10 50 Hertz
                                 /
```

```


                                       ;#*****************************************************************
                                       ;* .SLEFT - Input left hand switches
                                       ;#*****************************************************************

002254' 261 17 0 00 000000             .SLEFT: RPUT    (0,1,2,3,4,5,6)          ; save AC's

002263' 554 00 0 00 006113'            .SLF1:  HLRZ    CONSWW                   ; print out left hand switches as
002264' 037 01 0 00 030242                     PCRLF                            ;   they currently are
002265' 037 17 0 00 000003                     PNTOCF
002266' 037 01 0 00 005325'                    FMSG    <  LH Switches <# or ?> - >
002267' 260 17 0 00 002466'                    GO      .SWIN                    ; go input switches
002270' 254 00 0 00 002274'                    JRST    .SLF2                    ; typed a ?
002271' 254 00 0 00 002263'                    JRST    .SLF1                    ; error - re-ask question
002272' 506 01 0 00 006113'                    HRLM    1,CONSWW                 ; save left hand switches
002273' 254 00 0 00 002345'                    JRST    .SLFX                    ; exit

002274' 037 01 0 00 030242             .SLF2:  PCRLF
002275' 037 01 0 00 005333'                    PNTMSF  [ASCIZ /Type  Y,N,^,CR,^Z/]
002276' 037 01 0 00 030243                     PCRL2F
002277' 201 05 0 00 000001                     MOVEI   5,1                      ; init pointer to switch to ask
002300' 275 05 0 00 000002             .SLF3:  SUBI    5,2                      ; back up
002301' 350 00 0 00 000005             .SLF4:  AOS     5                        ; forward
002302' 335 00 0 00 000005                     SKIPGE  5                        ; backed up too much?
002303' 400 05 0 00 000000                     SETZ    5,                       ; yes - init pointer to first switch
002304' 303 05 0 00 000015                     CAILE   5,^D13                   ; reach end yet?
002305' 254 00 0 00 002345'                    JRST    .SLFX                    ; yes - exit

002306' 200 04 0 05 002527'            .SLF5:  MOVE    4,.LSWTN(5)              ; get proper bit in AC4
002307' 200 00 0 00 006113'                    MOVE    CONSWW                   ; get switches
002310' 616 00 0 00 000004                     TDNN    4
002311' 037 01 0 00 005337'                    FMSG    <N >
002312' 612 00 0 00 000004                     TDNE    4
002313' 037 01 0 00 005340'                    FMSG    <Y >
002314' 200 00 0 05 002511'                    MOVE    .LSWIT(5)                ; get SIXBIT switch name
002315' 037 01 0 00 000002                     PNTSXF                           ; print it
002316' 037 00 0 00 005341'                    TMSG    < - >
002317' 037 11 0 00 000003                     TTYINP
002320' 306 00 0 00 000015                     CAIN    15                       ; carraige return?
002321' 260 17 0 00 000045'                    GO      CLRBUF                   ; clear input buffer
002322' 306 00 0 00 000015                     CAIN    15                       ; carraige return?
002323' 254 00 0 00 002301'                    JRST    .SLF4                    ; yes - next switch
002324' 037 01 0 00 030242                     PCRLF
002325' 306 00 0 00 000032                     CAIN    32                       ; Control-Z?
002326' 254 00 0 00 002345'                    JRST    .SLFX                    ; yes - exit
002327' 306 00 0 00 000136                     CAIN    136                      ; ^ back up?
002330' 254 00 0 00 002300'                    JRST    .SLF3                    ; yes - go back up
002331' 306 00 0 00 000131                     CAIN    "Y"                      ; Y?
002332' 254 00 0 00 002341'                    JRST    .SLF6                    ; yes - set switch
002333' 302 00 0 00 000116                     CAIE    "N"                      ; N?
002334' 254 00 0 00 002306'                    JRST    .SLF5                    ; no - unrecognizable - reask question
```

```
002335' 200 00 0 00 006113'            MOVE    CONSWW                  ; get switches
002336' 412 00 0 00 000004             ANDCAM  4                       ; clear switch
002337' 202 00 0 00 006113'            MOVEM   CONSWW                  ; save switches
002340' 254 00 0 00 002301'            JRST    .SLF4                   ; continue
002341' 200 00 0 00 006113'    .SLF6:  MOVE    CONSWW                  ; get switches
002342' 434 00 0 00 000004             IOR     4                       ; set switch
002343' 202 00 0 00 006113'            MOVEM   CONSWW                  ; save switches
002344' 254 00 0 00 002301'            JRST    .SLF4                   ; continue

002345' 262 17 0 00 000006     .SLFX:  RGET    (6,5,4,3,2,1,0)         ; restore AC's
002354' 263 17 0 00 000000             RTN                             ; return


                               ;#*******************************************************************
                               ;* .SRIGT - Enter right hand switches
                               ;#*******************************************************************

002355' 261 17 0 00 000000     .SRIGT: RPUT    (0,1,2,3,4,5,6)         ; save AC's

002364' 550 00 0 00 006113'    .SRG2:  HRRZ    CONSWW                  ; print out right hand switches as
002365' 037 01 0 00 030242             PCRLF                           ;   they currently are
002366' 037 17 0 00 000003             PNTOCF
002367' 037 01 0 00 005342'            FMSG    <  RH Switches <# or ?> - >
002370' 260 17 0 00 002466'            GO      .SWIN                   ; go input switches
002371' 254 00 0 00 002375'            JRST    .SRG3                   ; typed a ?
002372' 254 00 0 00 002364'            JRST    .SRG2                   ; error - re-ask question
002373' 542 01 0 00 006113'            HRRM    1,CONSWW                        ; save right hand switches
002374' 254 00 0 00 002456'            JRST    .SRGX                   ; exit this switch handling

002375' 037 01 0 00 005333'    .SRG3:  PNTMSF  [ASCIZ /Type  Y,N,^,CR,^Z/]
002376' 037 01 0 00 030243             PCRL2F
002377' 201 05 0 00 000001             MOVEI   5,1                     ; init pointer to switch to ask
002400' 275 05 0 00 000002     .SRG4:  SUBI    5,2                     ; back up
002401' 350 00 0 00 000005     .SRG5:  AOS     5                       ; forward
002402' 335 00 0 00 000005             SKIPGE  5                       ; backed up too much?
002403' 400 05 0 00 000000             SETZ    5,                      ; yes - init pointer to first switch
002404' 303 05 0 00 000021             CAILE   5,^D17                  ; reach end yet?
002405' 254 00 0 00 002456'            JRST    .SRGX                   ; yes
002406' 336 00 0 05 002545'            SKIPN   .RSWIT(5)               ; switch zero?
002407' 254 00 0 00 002401'            JRST    .SRG5                   ; yes - go handle next one
```

```
002410' 201 04 0 00 400000      .SRG6:  MOVEI   4,400000                ; get a bit set in AC4
002411' 210 06 0 00 000005              MOVN    6,5                     ; negate for shifting
002412' 242 04 0 06 000000              LSH     4,(6)                   ; shift over properly
002413' 200 00 0 00 006113'             MOVE    CONSWW                  ; get switches
002414' 616 00 0 00 000004              TDNN    4
002415' 037 01 0 00 005337'             FMSG    <N >
002416' 612 00 0 00 000004              TDNE    4
002417' 037 01 0 00 005340'             FMSG    <Y >
002420' 200 00 0 05 002545'             MOVE    .RSWIT(5)               ; get SIXBIT switch name
002421' 037 01 0 00 000002              PNTSXF                          ; print it
002422' 037 00 0 00 005341'             TMSG    < - >
002423' 037 11 0 00 000003              TTYINP
002424' 306 00 0 00 000015              CAIN    15                      ; carraige return?
002425' 260 17 0 00 000045'             GO      CLRBUF                  ; clear input buffer
002426' 306 00 0 00 000015              CAIN    15                      ; carraige return?
002427' 254 00 0 00 002401'             JRST    .SRG5                   ; yes - next switch
002430' 037 01 0 00 030242              PCRLF
002431' 306 00 0 00 000032              CAIN    32                      ; Control-Z?
002432' 254 00 0 00 002456'             JRST    .SRGX                   ; yes - exit
002433' 302 00 0 00 000136              CAIE    136                     ; ^ back up?
002434' 254 00 0 00 002442'             JRST    .SRG6B                  ; no - continue
002435' 322 05 0 00 002410'             JUMPE   5,.SRG6                 ; at first switch? - yes - reask
002436' 370 00 0 00 000005      .SRG6A: SOS     5                       ; decrement switch
002437' 336 00 0 05 002545'             SKIPN   .RSWIT(5)               ; this switch exists?
002440' 254 00 0 00 002436'             JRST    .SRG6A                  ; no - try previous one
002441' 254 00 0 00 002410'             JRST    .SRG6                   ; yes - now continue

002442' 306 00 0 00 000131      .SRG6B: CAIN    "Y"                     ; Y?
002443' 254 00 0 00 002452'             JRST    .SRG7                   ; yes - set switch
002444' 302 00 0 00 000116              CAIE    "N"                     ; N?
002445' 254 00 0 00 002410'             JRST    .SRG6                   ; no - unrecognizable - reask question
002446' 200 00 0 00 006113'             MOVE    CONSWW                  ; get switches
002447' 620 00 0 04 000000              TRZ     (4)                     ; clear switch
002450' 202 00 0 00 006113'             MOVEM   CONSWW                  ; save switches
002451' 254 00 0 00 002401'             JRST    .SRG5                   ; continue
002452' 200 00 0 00 006113'     .SRG7:  MOVE    CONSWW                  ; get switches
002453' 660 00 0 04 000000              TRO     (4)                     ; set switch
002454' 202 00 0 00 006113'             MOVEM   CONSWW                  ; save switches
002455' 254 00 0 00 002401'             JRST    .SRG5                   ; continue

002456' 262 17 0 00 000006      .SRGX:  RGET    (6,5,4,3,2,1,0)         ; restore AC's
002465' 263 17 0 00 000000              RTN                             ; return
```

```
                                   ;#*********************************************************************
                                   ; .SWIN - Input switches (either as # or ?)
                                   ;
                                   ; Return +1 - ? typed
                                   ;        +2 - error
                                   ;        +3 - # typed (in AC1)
                                   ;        +4 - no change
                                   ;#*********************************************************************

002466' 037 11 0 00 000003         .SWIN:  TTYINP                                  ; get image mode character
002467' 306 00 0 00 000015                 CAIN    15                              ; CR typed?
                                           PJRST   [GO     CLRBUF                  ; clear input buffer
                                                    AOS     (P)                    ; increment return
                                                    AOS     (P)                    ; increment return
                                                    AOS     (P)                    ; increment return
                                                    PCRLF
002470' 254 00 0 00 005350'                         RTN]
002471' 306 00 0 00 000077                 CAIN    "?"                             ; ? typed?
                                           PJRST   [PCRL2F                         ; yes - exit
002472' 254 00 0 00 005356'                         RTN]
002473' 400 01 0 00 000000                 SETZ    1,                              ; clear number for starters
002474' 301 00 0 00 000060         .SWIN0: CAIL    "0"                             ; number typed is in a valid
002475' 303 00 0 00 000067                 CAILE   "7"                             ;   range?
                                           PJRST   [AOS    (P)                     ; no - error return
                                                    PCRLF
002476' 254 00 0 00 005353'                         RTN]
002477' 275 00 0 00 000060                 SUBI    60                              ; normalize to 0..7
002500' 242 01 0 00 000003                 LSH     1,3                             ; shift over current copy of number
002501' 270 01 0 00 000000                 ADD     1,0                             ; add in new digit
002502' 037 11 0 00 000003                 TTYINP                                  ; get a new character
002503' 302 00 0 00 000015                 CAIE    15                              ; carraige return?
002504' 254 00 0 00 002474'                JRST    .SWIN0                          ; no - go decode as a digit
002505' 260 17 0 00 000045'                GO      CLRBUF                          ; clear out extra
002506' 350 00 0 17 000000                 AOS     (P)                             ; set up return+2
002507' 350 00 0 17 000000                 AOS     (P)
002510' 263 17 0 00 000000                 RTN                                     ; return
```

```



                                ;#*********************************************      ******************
                                ; .LSWIT - Table of left hand switch names
                                ;#*********************************************        ****************

002511' 41 42 57 62 64 00       .LSWIT: SIXBIT  /ABORT /
002512' 62 63 64 41 62 64               SIXBIT  /RSTART/
002513' 64 57 64 41 54 63               SIXBIT  /TOTALS/
002514' 56 57 60 56 64 00               SIXBIT  /NOPNT /
002515' 60 56 64 54 60 64               SIXBIT  /PNTLPT/
002516' 44 51 56 47 00 00               SIXBIT  /DING  /
002517' 54 57 57 60 45 62               SIXBIT  /LOOPER/
002520' 45 62 63 64 57 60               SIXBIT  /ERSTOP/
002521' 60 41 54 45 62 63               SIXBIT  /PALERS/
002522' 62 45 54 51 41 42               SIXBIT  /RELIAB/
002523' 64 70 64 51 56 50               SIXBIT  /TXTINH/
002524' 55 57 44 44 66 43               SIXBIT  /MODDVC/
002525' 57 60 62 63 45 54               SIXBIT  /OPRSEL/
002526' 43 50 41 51 56 00               SIXBIT  /CHAIN /

002527' 400000  000000          .LSWTN: 400000,,0                       ; ABORT
002530' 200000  000000                  200000,,0                       ; RSTART
002531' 100000  000000                  100000,,0                       ; TOTALS
002532' 040000  000000                  040000,,0                       ; NOPNT
002533' 020000  000000                  020000,,0                       ; PNTLPT
002534' 010000  000000                  010000,,0                       ; DING
002535' 004000  000000                  004000,,0                       ; LOOPER
002536' 002000  000000                  002000,,0                       ; ERSTOP
002537' 001000  000000                  001000,,0                       ; PALERS
002540' 000400  000000                  000400,,0                       ; RELIAB
002541' 000200  000000                  000200,,0                       ; TXTINH
002542' 000040  000000                  000040,,0                       ; MODDVC
002543' 000010  000000                  000010,,0                       ; OPRSEL
002544' 000004  000000                  000004,,0                       ; CHAIN
```

```

                                      ;#*****************************************************************
                                      ; .RSWIT - Table of right hand switch names
                                      ;#*****************************************************************

                                      ; Here is the switch prompt table of SIXBIT switch names to use

  002545' 64 62 41 43 45 00           .RSWIT: SIXBIT  /TRACE/                 ; switch 18
  002546' 51 56 50 46 54 64                   SIXBIT  /INHFLT/                ; switch 19
  002547' 51 56 50 55 63 47                   SIXBIT  /INHMSG/                ; switch 20
  002550' 54 57 57 60 47 55                   SIXBIT  /LOOPGM/                ; switch 21
  002551' 54 57 57 60 64 63                   SIXBIT  /LOOPTS/                ; switch 22
  002552' 62 65 56 41 54 54                   SIXBIT  /RUNALL/                ; switch 23
  002553' 44 63 60 45 41 62                   SIXBIT  /DSPEAR/                ; switch 24
  002554' 65 44 45 42 65 47                   SIXBIT  /UDEBUG/                ; switch 25
  002555' 55 44 45 42 65 47                   SIXBIT  /MDEBUG/                ; switch 26
  002556' 63 44 45 42 65 47                   SIXBIT  /SDEBUG/                ; switch 27
  002557' 54 44 45 42 65 47                   SIXBIT  /LDEBUG/                ; switch 28
  002560' 51 44 45 42 65 47                   SIXBIT  /IDEBUG/                ; switch 29
  002561' 56 51 60 57 62 64                   SIXBIT  /NIPORT/                ; switch 30
  002562' 43 51 60 57 62 64                   SIXBIT  /CIPORT/                ; switch 31
  002563' 65 56 65 63 45 44                   SIXBIT  /UNUSED/                ; switch 32
  002564' 65 56 65 63 45 44                   SIXBIT  /UNUSED/                ; switch 33
  002565' 55 55 60 62 57 43                   SIXBIT  /MMPROC/                ; switch 34
  002566' 55 43 42 65 63 00                   SIXBIT  /MCBUS/                 ; switch 35



                                      ;#*****************************************************************
                                      ;* SWCHPT - Prints current state of switches
                                      ;#*****************************************************************

  002567' 261 17 0 00 000000          SWCHPT: PUT     0
  002570' 037 00 0 00 005360'                 TMSG    <Switches - >
  002571' 200 00 0 00 030046                  MOVE    CONSW
  002572' 037 13 0 00 000001                  PNTHWF
  002573' 037 00 0 00 005363'                 TMSGC   <>
  002574' 262 17 0 00 000000                  GET     0
  002575' 263 17 0 00 000000                  RTN
```

```


                                          ;#*********************************************************************
                                          ;* SWCOM - Switch complement routine (AC1 has switch command number 0-n)
                                          ;#*********************************************************************

002576' 261 17 0 00 000000               SWCOM:  RPUT    (0,1,2)                         ; save AC's
002601' 200 02 0 01 002666'                      MOVE    2,SWCOMP(1)                     ; get text
002602' 200 01 0 01 002635'                      MOVE    1,SWCOMT(1)                     ; get switch in octal
002603' 336 00 0 00 030037                       SKIPN   USER                            ; user mode?
002604' 607 01 0 00 777777                       TLNN    1,777777                        ; no - left hand switch set?
002605' 254 00 0 00 002615'                      JRST    SWCOMO                          ; no - continue

                                          ; Cannot complement

002606' 037 01 0 00 005364'                      FMSGC   <[Alter >
002607' 256 00 0 00 000002                       XCT     2
002610' 037 01 0 00 005366'                      FMSG    < with PDP11 Switch >
002611' 200 01 0 17 777777                       MOVE    1,-1(P)                         ; get switch number
                                                 XCT     [FMSG    <12>                   ; print it
                                                           FMSG    <11>
                                                           FMSG    <10>
                                                           FMSG    <9>
                                                           FMSG    <8>
                                                           FMSG    <7>
                                                           FMSG    <6>
                                                           FMSG    <5>
002612' 256 00 0 01 005403'                                FMSG    <1>](1)
002613' 037 01 0 00 005414'                      FMSGD   < only]>
002614' 254 00 0 00 002631'                      JRST    SWCOMX                          ; exit

                                          ; Ok to complement

002615' 260 17 0 00 001571'              SWCOMO: GO      SWITT                           ; get switches
002616' 430 00 0 00 000001                       XOR     1                               ; complement as specified
002617' 202 00 0 00 030046                       MOVEM   CONSW                           ; save switches
002620' 552 00 0 00 001576'                      HRRZM   SWRGT                           ; save right hand switches
002621' 037 01 0 00 005416'                      FMSGC   <[>
002622' 256 00 0 00 000002                       XCT     2                               ; print text
002623' 260 17 0 00 001571'                      GO      SWITT                           ; get switches
002624' 404 00 0 00 000001                       AND     1                               ; get only this switch
002625' 332 00 0 00 000000                       SKIPE                                   ; did it get set?
002626' 037 01 0 00 005417'                      FMSGD   < On]>                          ; yes
002627' 336 00 0 00 000000                       SKIPN                                   ; did it get set?
002630' 037 01 0 00 005421'                      FMSGD   < Off]>                         ; no
002631' 262 17 0 00 000002               SWCOMX: RGET    (2,1,0)                         ; restore AC's
002634' 263 17 0 00 000000                       RTN                                     ; continue
```

```
 1764
 1765                                    ; Switches in octal
 1766
 1767  002635' 040000  000000            SWCOMT: 040000,,0                           ; NOPNT
 1768  002636' 020000  000000                    020000,,0                           ; PNTLPT
 1769  002637' 010000  000000                    010000,,0                           ; DING
 1770  002640' 004000  000000                    004000,,0                           ; LOOPER
 1771  002641' 002000  000000                    002000,,0                           ; ERSTOP
 1772  002642' 001000  000000                    001000,,0                           ; PALERS
 1773  002643' 000400  000000                    000400,,0                           ; RELIAB
 1774  002644' 000200  000000                    000200,,0                           ; TXTINH
 1775  002645' 000010  000000                    000010,,0                           ; OPRSEL
 1776  002646' 000000  400000                    0,,400000                           ; TRACE
 1777  002647' 000000  200000                    0,,200000                           ; INHFLT
 1778  002650' 000000  100000                    0,,100000                           ; INHMSG
 1779  002651' 000000  040000                    0,,040000                           ; LOOPGM
 1780  002652' 000000  020000                    0,,020000                           ; LOOPTS
 1781  002653' 000000  010000                    0,,010000                           ; RUNALL
 1782  002654' 000000  004000                    0,,004000                           ; DSPEAR
 1783  002655' 000000  002000                    0,,002000                           ; UDEBUG
 1784  002656' 000000  001000                    0,,001000                           ; MDEBUG
 1785  002657' 000000  000400                    0,,000400                           ; SDEBUG
 1786  002660' 000000  000200                    0,,000200                           ; LDEBUG
 1787  002661' 000000  000100                    0,,000100                           ; IDEBUG
 1788  002662' 000000  000040                    0,,000040                           ; NIPORT
 1789  002663' 000000  000020                    0,,000020                           ; CIPORT
 1790  002664' 000000  000002                    0,,000002                           ; MMPROC
 1791  002665' 000000  000001                    0,,000001                           ; MCBUS
```

```


                                        ; Switch text

002666' 037 01 0 00 005423'     SWCOMP: FMSG    <NOPNT>
002667' 037 01 0 00 005425'             FMSG    <PNTLPT>
002670' 037 01 0 00 005427'             FMSG    <DING>
002671' 037 01 0 00 005430'             FMSG    <LOOPER>
002672' 037 01 0 00 005432'             FMSG    <ERSTOP>
002673' 037 01 0 00 005434'             FMSG    <PALERS>
002674' 037 01 0 00 005436'             FMSG    <RELIAB>
002675' 037 01 0 00 005440'             FMSG    <TXTINH>
002676' 037 01 0 00 005442'             FMSG    <OPRSEL>
002677' 037 01 0 00 005444'             FMSG    <TRACE>
002700' 037 01 0 00 005446'             FMSG    <INHFLT>
002701' 037 01 0 00 005450'             FMSG    <INHMSG>
002702' 037 01 0 00 005452'             FMSG    <LOOPGM>
002703' 037 01 0 00 005454'             FMSG    <LOOPTS>
002704' 037 01 0 00 005456'             FMSG    <RUNALL>
002705' 037 01 0 00 005460'             FMSG    <DSPEAR>
002706' 037 01 0 00 005462'             FMSG    <UDEBUG>
002707' 037 01 0 00 005464'             FMSG    <MDEBUG>
002710' 037 01 0 00 005466'             FMSG    <SDEBUG>
002711' 037 01 0 00 005470'             FMSG    <LDEBUG>
002712' 037 01 0 00 005472'             FMSG    <IDEBUG>
002713' 037 01 0 00 005474'             FMSG    <NIPORT>
002714' 037 01 0 00 005476'             FMSG    <CIPORT>
002715' 037 01 0 00 005500'             FMSG    <MMPROC>
002716' 037 01 0 00 005502'             FMSG    <MCBUS>

```

```
                                                SUBTTL  Miscellaneous Routines

                                        ;#**********************************************************
                                        ;* PARSER - Additional CPU parity error service
                                        ;
                                        ; This routine is called by the subroutine package automatically when the
                                        ; processor is interrupted with a memory parity error.  It furnishes some
                                        ; additional useful information.  (A) IPA status (B) Channel  status.  If
                                        ; running on a KL10 in exec mode, and the parity error occurs not  in the
                                        ; program area but the  data  buffer, the  error will be cleared and  the
                                        ; diagnostic will be forced to loop on the error.
                                        ;#**********************************************************

002717' 332 00 0 00 030037              PARSER: SKIPE   USER                            ; user mode?
002720' 260 17 0 00 000000*                     GO      FFF                             ; yes - fatal error
002721' 261 17 0 00 000000                      RPUT    (0,1)                           ; save AC's
002723' 402 00 0 00 000001                      SETZM   1
002724' 260 17 0 00 000000*                     GO      RDCSR                           ; read CSR register
002725' 255 00 0 00 000000                       JFCL                                   ; error return
002726' 037 01 0 00 005504'                     FMSGC   <************* Memory Parity Error Occurred *************>
002727' 037 01 0 00 005520'                     FMSGC   <CSR:  >
002730' 200 00 0 00 000001                      MOVE    1
002731' 037 13 0 00 000001                      PNTHWF
002732' 260 17 0 00 000000*                     GO      GETLOG                          ; get logout area for channel
002733' 260 17 0 00 000000*                     GO      LOGPNT                          ; print logout data
002734' 336 00 0 00 030041                      SKIPN   KLFLG                           ; KL-PROCESSOR?
002735' 260 17 0 00 002720*                     GO      FFF                             ; no - fatal error
002736' 255 00 0 00 000000                      JFCL                                    ; yes
002737' 256 00 0 00 005522'                     XCT     [700400,,0]                     ; read error addr register
002740' 404 00 0 00 005523'                     AND     0,[17,777777]                   ; save 22 bit address
002741' 202 00 0 00 006141'                     MOVEM   XYZ#                            ; save
002742' 311 00 1 00 006112'                     CAML    0,@ABUFF#                       ; in buffer area
002743' 254 00 0 00 002746'                     JRST    CORECT                          ; yes. we can continue
002744' 037 01 0 00 005524'                     FMSGCD  <Error not in data buffer area, cannot continue>
002745' 260 17 0 00 002735*                     GO      FFF                             ; fatal out

002746' 037 01 0 00 005537'             CORECT: FMSGCD  <Error is in the data buffer, diag will continue>
002747' 256 00 0 00 005552'                     XCT     [700240,,0]                     ; CONI APR
002750' 554 00 0 00 000000                      HLRZ    0,0                             ; swaps flags to right
002751' 435 00 0 00 000001                      IORI    0,1                             ; include PI channel
002752' 405 00 0 00 007777                      ANDI    0,7777                          ; save FLAGS and PI
002753' 434 00 0 00 005553'                     IOR     0,[700200,,120000]              ; clr and re-enable
002754' 256 00 0 00 000000                      XCT     0                               ; does the CONO
002755' 256 00 0 00 005552'                     XCT     [700240,,0]                     ; another CONI
002756' 262 17 0 00 000001                      GET     1
002757' 262 17 0 00 000000                      GET     0
002760' 254 12 1 00 030104                      JEN     @ITRCH1                         ; dismiss the interrupt
```

```

                                        ;#****************************************************************
                                        ;* .COMM - Match a command entered against a command table.
                                        ;
                                        ; Handle erroneous input by returning to re-ask question.
                                        ;
                                        ; Arguments:    AC1 - Contains the SIXBIT command entered
                                        ;               AC2 - Contains address of command table
                                        ;
                                        ; Returns:      RTN+1 - Return error address (for re-asking question)
                                        ;               RTN+2 - Valid command recognized, offset in command
                                        ;                       table is returned in AC1
                                        ;
                                        ;#****************************************************************

002761' 261 17 0 00 000000              .COMM:  RPUT    (0,2,3)                         ; save some AC's
002764' 202 01 0 00 003006'                     MOVEM   1,.CTYP                         ; save command typed

                                        ; Now look through each command

002765' 474 03 0 00 000000                      SETO    3,                              ; init test check pointer
002766' 350 00 0 00 000003              .COMM0: AOS     3                               ; point to next command
002767' 260 17 0 00 003012'                     GO      .COMM2                          ; see if this is the test selected
002770' 254 00 0 00 002766'                     JRST    .COMM0                          ; no - try next command
002771' 254 00 0 00 002776'                     JRST    .COMM1                          ; yes - go handle

                                        ; Error exit

002772' 262 17 0 00 000003                      RGET    (3,2,0)                         ; restore AC's
002775' 263 17 0 00 000000                      RTN                                     ; return + 1 (error)

                                        ; Found a valid command

002776' 202 03 0 00 003005'             .COMM1: MOVEM   3,.CGOT                         ; save command number
002777' 202 03 0 00 000001                      MOVEM   3,1                             ; put into AC1
003000' 262 17 0 00 000003                      RGET    (3,2,0)                         ; restore AC's
003003' 350 00 0 17 000000                      AOS     (P)                             ; set up return + 2 (valid return)
003004' 263 17 0 00 000000                      RTN

                                        ; Storage areas

003005' 000000  000000                  .CGOT:  0                                       ; command number

003006' 000000  000000                  .CTYP:  0                                       ; command typed (SIXBIT)
003007' 000000  000000                          0

003010' 000000  000000                  .CTBL:  0                                       ; command to compare
003011' 000000  000000                          0
```

```

                                    ;#*********************************************************************
                                    ;* .COMM2 - Compare command typed to table to see if valid
                                    ;
                                    ; AC2 - Contains address of sixbit command table
                                    ; AC3 - Contains current command number (0..)
                                    ;#*********************************************************************

003012' 261 17 0 00 000000          .COMM2: RPUT    (0,1,2,3)                       ; save AC's
003016' 270 02 0 00 000003                  ADD     2,3                             ; calculate address of entry to compare
003017' 200 00 0 02 000000                  MOVE    (2)                             ; get cmd table entry
003020' 322 00 0 00 003047'                 JUMPE   .COMM6                          ; reached end of table? - yes - exit
003021' 202 00 0 00 003010'                 MOVEM   .CTBL                           ; save cmd table entry
003022' 200 02 0 00 005554'                 MOVE    2,[POINT 6,.CTYP,5]             ; set up byte pointers
003023' 200 03 0 00 005555'                 MOVE    3,[POINT 6,.CTBL,5]             ;   ...
003024' 135 00 0 00 000002          .COMM3: LDB     0,2                             ; get character of typed
003025' 135 01 0 00 000003                  LDB     1,3                             ; get character of table
003026' 322 00 0 00 003034'                 JUMPE   .COMM4                          ; end of typed? yes - done
003027' 312 00 0 00 000001                  CAME    1                               ; no - same as table?
003030' 254 00 0 00 003042'                 JRST    .COMM5                          ; no - no match - exit
003031' 133 00 0 00 000002                  IBP     2                               ; increment byte pointers
003032' 133 00 0 00 000003                  IBP     3                               ;   ...
003033' 254 00 0 00 003024'                 JRST    .COMM3                          ; end of table entry? no - keep looping
003034' 262 17 0 00 000003          .COMM4: RGET    (3,2,1,0)                       ; yes - match - done
003040' 350 00 0 17 000000                  AOS     (P)                             ; set up return+2
003041' 263 17 0 00 000000                  RTN                                     ; return+2
003042' 262 17 0 00 000003          .COMM5: RGET    (3,2,1,0)                       ; no match
003046' 263 17 0 00 000000                  RTN                                     ; return+1
003047' 262 17 0 00 000003          .COMM6: RGET    (3,2,1,0)                       ; return as end of list
003053' 350 00 0 17 000000                  AOS     (P)                             ; set up return+2
003054' 350 00 0 17 000000                  AOS     (P)                             ; set up return+3
003055' 263 17 0 00 000000                  RTN                                     ; return+3
```

```

                                ;#********************************************************************
                                ;* .SARG - Obtain SIXBIT argument
                                ;
                                ; Arguments:  ARGFLG - 0 - No more arguments
                                ;                     -1 - More arguments have been given
                                ;             AC1 - Contains executable help message
                                ;
                                ; Returns:  ARGUM - Contains SIXBIT argument
                                ;           ARGFLG - Updated
                                ;           +1 - Error occurred/Altmode typed/Question mark typed
                                ;           +2 - No argument
                                ;           +3 - Ok
                                ;#********************************************************************

003056' 261 17 0 00 000000      .SARG:  RPUT    (0,1,2,3,ALTMGO)        ; save AC's
                                        MOVEI   [SETOM  ALTF            ; set up altmode handling to exit
                                                 PCRL                   ;   +1 after setting altmode flag
003063' 201 00 0 00 005556'                      JRST   .SARX1]         ;   and skipping a line
003064' 202 00 0 00 030063              MOVEM   ALTMGO

                                ; Check if any arguments were given at all

003065' 336 00 0 00 003277'             SKIPN   ARGFLG                  ; any arguments?
003066' 254 00 0 00 003107'             JRST    .SARX2                  ; no - exit 'no argument'

                                ; Check for file input

003067' 332 00 0 00 000642*             SKIPE   FINPUT                  ; file input?
                                        JRST    [GO     FSARG           ; yes - input octal argument
                                                 JRST   .SARX1          ; error or EOF occurred
003070' 254 00 0 00 005561'                      JRST   .SARG1]         ; continue

                                ; Input SIXBIT argument

003071' 037 10 0 00 000003      .SARG0: TTSIXB                          ; get argument
003072' 007 00 0 00 003116'             ALTCHK  .SARGA                  ; error - go handle

                                ; Check result

                                .SARG1: JUMPE   [MOVE   $TTCHR          ; anything typed?  if a space, keep
                                                 CAIN   40              ;   looking for an argument, if not
                                                 JRST   .SARG0          ;   exit
003073' 322 00 0 00 005564'                      JRST   .SARX2]
003074' 202 00 0 00 003276'             MOVEM   ARGUM                   ; save argument
```

```

                                        ; Check for last argument

        003075' 402 00 0 00 003277'             SETZM   ARGFLG                          ; clear 'argument typed' flag
        003076' 200 01 0 00 030231              MOVE    1,$TTCHR                        ; get last character typed
        003077' 302 01 0 00 000015              CAIE    1,15                            ; CR? (end of line)
        003100' 476 00 0 00 003277'             SETOM   ARGFLG                          ; no - set 'argument typed' flag

                                        ; Check if this should have been the last argument

        003101' 331 00 0 17 777775              SKIPL   -3(P)                           ; help message negative?
        003102' 254 00 0 00 003106'             JRST    .+4                             ; no - exit
        003103' 260 17 0 00 003324'             GO      LASARG                          ; check for last argument
        003104' 254 00 0 00 003110'             JRST    .SARX1                          ; error - exit +1
                                                JRST    [MOVM   1,-3(P)                 ; any help here?
                                                         JUMPE  1,.SARX1                ; no - continue
                                                         XCT    1                       ; yes - execute it
        003105' 254 00 0 00 005570'                      JRST   .SARX1]                 ; yes - handle
        003106' 350 00 0 17 777773              AOS     -5(P)                           ; adjust return
        003107' 350 00 0 17 777773      .SARX2: AOS     -5(P)                           ; adjust return

                                        ; Exit

        003110' 262 17 0 00 030063      .SARX1: RGET    (ALTMGO,3,2,1,0)                ; restore AC's
        003115' 263 17 0 00 000000              RTN                                     ; return

                                        ; Altmode typed

        003116' 262 17 0 00 006140'     .SARGA: GET     XXT#                            ; adjust stack properly
        003117' 336 00 0 00 030225              SKIPN   $TWCNT                          ; did a timeout occur?
        003120' 254 00 0 00 003107'             JRST    .SARX2                          ; yes - assume 'no argument'
        003121' 200 01 0 00 030231              MOVE    1,$TTCHR                        ; no - get character typed
        003122' 306 01 0 00 000077              CAIN    1,77                            ; was a question mark typed?
                                                JRST    [MOVM   1,-3(P)                 ; yes - any help here?
                                                         JUMPE  1,.SARX1                ; no - exit
                                                         XCT    1                       ; yes - execute it
        003123' 254 00 0 00 005570'                      JRST   .SARX1]                 ; exit
        003124' 302 01 0 00 000040              CAIE    1,40                            ; space (delimiter character) typed?
                                                JRST    [FMSGCD <? Argument error> ; no - error - exit
        003125' 254 00 0 00 005601'                      JRST   .SARX1]
        003126' 254 00 0 00 003073'             JRST    .SARG1                          ; continue
```

```

                                        ;#*********************************************************************
                                        ;* .OARG - Obtain octal argument
                                        ;
                                        ; Arguments:  ARGFLG - 0 - No more arguments
                                        ;                     -1 - More arguments have been given
                                        ;             AC1 - Contains executable help message
                                        ;
                                        ; Returns:  ARGUM - Contains octal argument
                                        ;           ARGFLG - Updated
                                        ;
                                        ;           +1 - Error occurred/Altmode typed/Question mark typed
                                        ;           +2 - No argument
                                        ;           +3 - Ok
                                        ;#*********************************************************************

003127' 261 17 0 00 000000              .OARG:  RPUT    (0,1,2,3,ALTMGO)        ; save AC's
                                                MOVEI   [SETOM  ALTF            ; set up altmode handling to exit
                                                         PCRL                   ;   +1 after setting altmode flag
003134' 201 00 0 00 005603'                              JRST   .OARX1]         ;   and skipping a line
003135' 202 00 0 00 030063                      MOVEM   ALTMGO

                                        ; Check if any arguments were given at all

003136' 336 00 0 00 003277'                     SKIPN   ARGFLG                  ; any arguments?
003137' 254 00 0 00 003161'                     JRST    .OARX2                  ; no - exit 'no argument'

                                        ; Check for file input

003140' 332 00 0 00 003067*                     SKIPE   FINPUT                  ; file input?
                                                JRST    [GO     FOARG           ; yes - input octal argument
                                                         JRST   .OARX1          ; error or EOF occurred
003141' 254 00 0 00 005606'                              JRST   .OARG1]         ; continue

                                        ; Input octal argument

003142' 037 03 0 00 000003              .OARG0: TTIOCT                          ; get a number
003143' 007 00 0 00 003170'                     ALTCHK  .OARGA                  ; error - go handle

                                        ; Check result

003144' 202 00 0 00 003276'             .OARG1: MOVEM   ARGUM                   ; save argument
003145' 336 00 0 00 030510                      SKIPN   TTNBRF                  ; any number typed?
                                                JRST    [MOVE   1,$TTCHR        ; anything typed?  if a space, keep
                                                         CAIN   1,40            ;   looking for an argument, if not
                                                         JRST   .OARG0          ;   exit
003146' 254 00 0 00 005611'                              JRST   .OARX2]
```

```

                                        ; Check for last argument

003147' 402 00 0 00 003277'                     SETZM   ARGFLG                  ; clear 'argument typed' flag
003150' 200 01 0 00 030231                      MOVE    1,$TTCHR                ; get last character typed
003151' 302 01 0 00 000015                      CAIE    1,15                    ; CR? (end of line)
003152' 476 00 0 00 003277'                     SETOM   ARGFLG                  ; no - set 'argument typed' flag

                                        ; Check if this should have been the last argument

003153' 331 00 0 17 777775                      SKIPL   -3(P)                   ; help message negative?
003154' 254 00 0 00 003160'                     JRST    .+4                     ; no - exit
003155' 260 17 0 00 003324'                     GO      LASARG                  ; check for last argument
003156' 254 00 0 00 003162'                     JRST    .OARX1                  ; error - exit +1
                                                JRST    [MOVM   1,-3(P)         ; any help here?
                                                         JUMPE  1,.OARX1        ; no - continue
                                                         XCT    1               ; yes - execute it
003157' 254 00 0 00 005615'                              JRST   .OARX1]         ; yes - handle
003160' 350 00 0 17 777773                      AOS     -5(P)                   ; adjust return
003161' 350 00 0 17 777773              .OARX2: AOS     -5(P)                   ; adjust return

                                        ; Exit

003162' 262 17 0 00 030063              .OARX1: RGET    (ALTMGO,3,2,1,0)        ; restore AC's
003167' 263 17 0 00 000000                      RTN                             ; return

                                        ; Altmode typed

003170' 262 17 0 00 006140'             .OARGA: GET     XXT#                    ; adjust stack properly
003171' 336 00 0 00 030225                      SKIPN   $TWCNT                  ; did a timeout occur?
003172' 254 00 0 00 003161'                     JRST    .OARX2                  ; yes - assume 'no argument'
003173' 200 01 0 00 030231                      MOVE    1,$TTCHR                ; no - get character typed
003174' 306 01 0 00 000077                      CAIN    1,77                    ; was a question mark typed?
                                                JRST    [MOVM   1,-3(P)         ; yes - any help here?
                                                         JUMPE  1,.OARX1        ; no - exit
                                                         XCT    1               ; yes - execute it
003175' 254 00 0 00 005615'                              JRST   .OARX1]         ; exit
003176' 302 01 0 00 000040                      CAIE    1,40                    ; space (delimiter character) typed?
                                                JRST    [FMSGCD <? Argument error> ; no - error - exit
003177' 254 00 0 00 005621'                              JRST   .OARX1]
003200' 200 00 0 00 030233                      MOVE    $TYPNB                  ; get number typed
003201' 254 00 0 00 003144'                     JRST    .OARG1                  ;   and exit
```

```

                                        ;#*********************************************************************
                                        ;* DECYN - Decode an argument which should be NO or YES
                                        ;
                                        ; Arguments:  ARGUM - Contains sixbit argument
                                        ;
                                        ; Return:     +1 - Unrecognizable (print message also)
                                        ;             +2 - No
                                        ;             +3 - Yes
                                        ;#*********************************************************************

 003202' 261 17 0 00 000000             DECYN:  RPUT    (0,1,2)                 ; save AC's
 003205' 200 01 0 00 003276'                    MOVE    1,ARGUM                 ; get command into AC1
 003206' 201 02 0 00 003220'                    MOVEI   2,DECLIS                ; get address of cmd list
 003207' 260 17 0 00 002761'                    GO      .COMM                   ; handle command decoding
                                                 JRST   [FMSGCD <? Unrecognized argument>
 003210' 254 00 0 00 005631'                             JRST   DECERR]
 003211' 332 00 0 00 000001                     SKIPE   1                       ; NO typed?
 003212' 350 00 0 17 777775                     AOS     -3(P)           ; increment return
 003213' 350 00 0 17 777775                     AOS     -3(P)           ; increment return
 003214' 262 17 0 00 000002             DECERR: RGET    (2,1,0)         ; restore AC's
 003217' 263 17 0 00 000000                     RTN                     ; return

 003220' 56 57 00 00 00 00              DECLIS: SIXBIT  /NO/
 003221' 71 45 63 00 00 00                      SIXBIT  /YES/
 003222' 000 00 0 00 000000                     Z
```

```

                                ;#*********************************************************************
                                ;* .DARG - Obtain octal/decimal argument
                                ;
                                ; Arguments:  ARGFLG - 0 - No more arguments
                                ;                     -1 - More arguments have been given
                                ;             AC1 - Contains executable help message
                                ;
                                ; Returns:  ARGUM - Contains octal argument
                                ;           ARGFLG - Updated
                                ;
                                ;           +1 - Error occurred/Altmode typed/Question mark typed
                                ;           +2 - No argument
                                ;           +3 - Ok
                                ;#*********************************************************************

003223' 261 17 0 00 000000      .DARG:  RPUT    (0,1,2,3,ALTMGO)        ; save AC's
                                        MOVEI   [SETOM  ALTF            ; set up altmode handling to exit
                                                 PCRL                   ;   +1 after setting altmode flag
003230' 201 00 0 00 005633'                      JRST   .DARX1]         ;   and skipping a line
003231' 202 00 0 00 030063              MOVEM   ALTMGO

                                ; Check if any arguments were given at all

003232' 336 00 0 00 003277'             SKIPN   ARGFLG                  ; any arguments?
003233' 254 00 0 00 003255'             JRST    .DARX2                  ; no - exit 'no argument'

                                ; Check for file input

003234' 332 00 0 00 003140*             SKIPE   FINPUT                  ; file input?
                                        JRST    [GO     FOARG           ; yes - input octal argument
                                                 JRST   .DARX1          ; error or EOF occurred
003235' 254 00 0 00 005636'                      JRST   .DARG1]         ; continue

                                ; Input octal/decimal argument

003236' 037 05 0 00 000003      .DARG0: TTICNV                          ; get a number
003237' 007 00 0 00 003264'             ALTCHK  .DARGA                  ; error - go handle

                                ; Check result

003240' 202 00 0 00 003276'     .DARG1: MOVEM   ARGUM                   ; save argument
003241' 336 00 0 00 030510              SKIPN   TTNBRF                  ; any number typed?
                                        JRST    [MOVE   1,$TTCHR        ; anything typed?  if a space, keep
                                                 CAIN   1,40            ;   looking for an argument, if not
                                                 JRST   .DARG0          ;   exit
003242' 254 00 0 00 005641'                      JRST   .DARX2]
```

```

                                        ; Check for last argument

003243' 402 00 0 00 003277'                     SETZM   ARGFLG                          ; clear 'argument typed' flag
003244' 200 01 0 00 030231                      MOVE    1,$TTCHR                        ; get last character typed
003245' 302 01 0 00 000015                      CAIE    1,15                            ; CR? (end of line)
003246' 476 00 0 00 003277'                     SETOM   ARGFLG                          ; no - set 'argument typed' flag

                                        ; Check if this should have been the last argument

003247' 331 00 0 17 777775                      SKIPL   -3(P)                           ; help message negative?
003250' 254 00 0 00 003254'                     JRST    .+4                             ; no - exit
003251' 260 17 0 00 003324'                     GO      LASARG                          ; check for last argument
003252' 254 00 0 00 003256'                     JRST    .DARX1                          ; error - exit +1
                                                JRST    [MOVM   1,-3(P)                 ; any help here?
                                                         JUMPE  1,.DARX1                ; no - continue
                                                         XCT    1                       ; yes - execute it
003253' 254 00 0 00 005645'                              JRST   .DARX1]                 ; yes - handle
003254' 350 00 0 17 777773                      AOS     -5(P)                           ; adjust return
003255' 350 00 0 17 777773              .DARX2: AOS     -5(P)                           ; adjust return

                                        ; Exit

003256' 262 17 0 00 030063              .DARX1: RGET    (ALTMGO,3,2,1,0)                ; restore AC's
003263' 263 17 0 00 000000                      RTN                                     ; return

003264' 262 17 0 00 006140'             .DARGA: GET     XXT#                            ; adjust stack properly
003265' 336 00 0 00 030225                      SKIPN   $TWCNT                          ; did a timeout occur?
003266' 254 00 0 00 003255'                     JRST    .DARX2                          ; yes - assume 'no argument'
003267' 200 01 0 00 030231                      MOVE    1,$TTCHR                        ; no - get character typed
003270' 306 01 0 00 000077                      CAIN    1,77                            ; was a question mark typed?
                                                JRST    [MOVM   1,-3(P)                 ; yes - any help here?
                                                         JUMPE  1,.DARX1                ; no - exit
                                                         XCT    1                       ; yes - execute it
003271' 254 00 0 00 005645'                              JRST   .DARX1]                 ; exit
003272' 302 01 0 00 000040                      CAIE    1,40                            ; space (delimiter character) typed?
                                                JRST    [FMSGCD <? Argument error> ; no - error - exit
003273' 254 00 0 00 005651'                              JRST   .DARX1]
003274' 200 00 0 00 030233                      MOVE    $TYPNB                          ; get number typed
003275' 254 00 0 00 003240'                     JRST    .DARG1                          ;   and exit

                                        ; Argument variables

003276' 000000  000000                  ARGUM:  0
003277' 000000  000000                  ARGFLG: 0
```

```

                                    ;#*******************************************************************
                                    ;* CHKARG - Check for argument when there shouldn't be any
                                    ;
                                    ; Arguments:     AC1 - Contains confirmation message
                                    ;
                                    ; Return:        +1 - An argument was typed
                                    ;                +2 - No argument was typed
                                    ;#*******************************************************************

003300' 261 17 0 00 000000          CHKARG: RPUT    (0,1,ALTMGO)            ; save AC's
003303' 336 00 0 00 003277'                 SKIPN   ARGFLG                  ; any argument at all?
003304' 254 00 0 00 003317'                 JRST    CHKARX                  ; no - exit +2
                                            MOVEI   [SETOM  ALTF            ; set up altmode handling to exit
                                                     PCRL                   ;   +1 after setting altmode flag
003305' 201 00 0 00 005653'                          JRST   CHKARA]         ;   and skipping a line
003306' 202 00 0 00 030063                  MOVEM   ALTMGO

                                    ; Input argument to see if ? typed

003307' 332 00 0 00 003234*                 SKIPE   FINPUT                  ; file input switch set?
                                            JRST    [GO     FEOL            ; yes - move to end of line
003310' 254 00 0 00 005656'                          JRST   CHKARX]
003311' 037 05 0 00 000003          CHKAR0: TTICNV                          ; get a number
                                            ALTCHK  [GET    XXT#            ; adjust stack properly
                                                     SKIPN  $TWCNT          ; did a timeout occur?
                                                     JRST   CHKARY          ; yes - error
                                                     MOVE   1,$TTCHR        ; no - get character typed
                                                     CAIE   1,77            ; was a question mark typed?
                                                     JRST   .+1             ; no - error
                                                     XCT    -1(P)           ; print help message
003312' 007 00 0 00 005660'                          JRST   CHKARA]         ;   and exit
003313' 336 00 0 00 030510                  SKIPN   TTNBRF                  ; any number typed?
                                            JRST    [MOVE   $TTCHR          ; anything typed?  if a space, keep
                                                     CAIN   40              ;   looking for an argument, if not
                                                     JRST   CHKAR0          ;   exit
                                                     CAIN   15
                                                     JRST   CHKARX
003314' 254 00 0 00 005670'                          JRST   CHKARY]

003315' 037 01 0 00 005676'         CHKARY: FMSGCD  <? No argument expected>
003316' 334 00 0 00 000000                  SKIPA
003317' 350 00 0 17 777775          CHKARX: AOS     -3(P)                   ; adjust stack
003320' 262 17 0 00 030063          CHKARA: RGET    (ALTMGO,1,0)            ; restore AC's
003323' 263 17 0 00 000000                  RTN                             ; return
```

```

                                        ;#*********************************************************************
                                        ;* LASARG - Check for last argument when there shouldn't be any more
                                        ;
                                        ; Return:       +1 - Error - more arguments
                                        ;               +2 - Question mark typed
                                        ;               +3 - No more arguments typed
                                        ;#*********************************************************************

003324' 261 17 0 00 000000      LASARG: RPUT    (0,1,ALTMGO)                    ; save AC's
003327' 336 00 0 00 003277'             SKIPN   ARGFLG                          ; any argument at all?
003330' 254 00 0 00 003343'             JRST    LASARX                          ; no - exit +3
                                        MOVEI   [SETOM  ALTF                    ; set up altmode handling to exit
                                                 PCRL                           ;   +1 after setting altmode flag
003331' 201 00 0 00 005704'                      JRST   LASARA]                 ;   and skipping a line
003332' 202 00 0 00 030063              MOVEM   ALTMGO
003333' 332 00 0 00 003307*             SKIPE   FINPUT                          ; file input switch set?
                                        JRST    [GO     FEOL                    ; yes - move to end of line
003334' 254 00 0 00 005707'                      JRST   LASARX]
003335' 037 05 0 00 000003      LASAR0: TTICNV                                  ; get a number
                                        ALTCHK  [GET    XXT#                    ; adjust stack properly
                                                 SKIPN  $TWCNT                  ; did a timeout occur?
                                                 JRST   LASAR1                  ; yes - error
                                                 MOVE   1,$TTCHR                ; get character typed
                                                 CAIE   1,77                    ; question mark?
                                                 JRST   .+1                     ; yes - continue
003336' 007 00 0 00 005711'                      JRST   LASARX+1]               ; no - handle
003337' 336 00 0 00 030510              SKIPN   TTNBRF                          ; any number typed?
                                        JRST    [MOVE   $TTCHR                  ; anything typed?  if a space, keep
                                                 CAIN   40                      ;   looking for an argument, if not
                                                 JRST   LASAR0                  ;   exit
                                                 CAIN   15
                                                 JRST   LASARX
003340' 254 00 0 00 005720'                      JRST   LASAR1]

003341' 037 01 0 00 005726'     LASAR1: FMSGCD  <? Too many arguments>
003342' 254 00 0 00 003345'             JRST    .+3
003343' 350 00 0 17 777775      LASARX: AOS     -3(P)                           ; adjust stack
003344' 350 00 0 17 777775              AOS     -3(P)
003345' 262 17 0 00 030063      LASARA: RGET    (ALTMGO,1,0)                    ; restore AC's
003350' 263 17 0 00 000000              RTN                                     ; return
```

```


                                 ;#********************************************************************
                                 ;* FIOFF - Routine to turn off the 'file input' switch
                                 ;#********************************************************************

003351' 336 00 0 00 003333*      FIOFF:  SKIPN   FINPUT                  ; switch set?
003352' 263 17 0 00 000000               RTN                             ; no - return
003353' 261 17 0 00 000000               PUT     0                       ; yes - save AC0
003354' 037 01 0 00 005733'              FMSGC   <[Input from >          ; print message
003355' 200 00 0 00 000634*              MOVE    TAKFIL                  ; print file name
003356' 037 01 0 00 000002               PNTSXF
003357' 037 01 0 00 000056               PNTCIF  "."
003360' 200 00 0 00 000000#              MOVE    TAKFIL+1
003361' 037 01 0 00 000002               PNTSXF
003362' 037 01 0 00 005736'              FMSGD   < aborted]>
003363' 402 00 0 00 003351*              SETZM   FINPUT                  ; clear 'file input' flag
003364' 262 17 0 00 000000               GET     0                       ; restore AC0
003365' 263 17 0 00 000000               RTN                             ; return
```

```

                                  ;#*********************************************************************
                                  ;* FINCMD - Input a sixbit command from a file
                                  ;
                                  ; Return +1 - Error/EOF
                                  ;        +2 - Sixbit word in AC0
                                  ;
                                  ;#*********************************************************************

003366' 261 17 0 00 000001        FINCMD: RPUT    (1,2)                   ; save AC's
003370' 402 00 0 00 006117'               SETZM   FINSIX#                 ; initialize word
003371' 400 01 0 00 000000                SETZ    1,                      ; byte count
003372' 200 02 0 00 005741'               MOVE    2,[POINT 6,FINSIX]      ; initial byte pointer

003373' 260 17 0 00 000621'       FINCMO: GO      CHRINS                  ; get a character
003374' 254 00 0 00 003422'                JRST   FINCMX                  ; exit - error or EOF occurred
003375' 306 00 0 00 000040                CAIN    " "                     ; space typed?
003376' 254 00 0 00 003412'               JRST    FINCMC                  ; yes - exit valid command in AC0
003377' 306 00 0 00 000054                CAIN    ","                     ; comma typed?
003400' 254 00 0 00 003412'               JRST    FINCMC                  ; yes - exit valid command in AC0
003401' 306 00 0 00 000015                CAIN    15                      ; carriage return typed?
003402' 254 00 0 00 003412'               JRST    FINCMC                  ; yes - exit valid command in AC0
003403' 350 00 0 00 000001                AOS     1                       ; increment byte count
003404' 303 00 0 00 000140                CAILE   140                     ; lowercase?
003405' 275 00 0 00 000040                SUBI    40                      ; yes - make into upper case
003406' 275 00 0 00 000040                SUBI    40                      ; convert to sixbit
003407' 307 01 0 00 000006                CAIG    1,6                     ; exceed 6 characters?
003410' 136 00 0 00 000002                IDPB    2                       ; no - save sixbit byte
003411' 254 00 0 00 003373'               JRST    FINCMO                  ; get next character

                                  ; Exit with command in AC0

003412' 202 00 0 00 030231        FINCMC: MOVEM   $TTCHR                  ; save last character typed
003413' 306 00 0 00 000015                CAIN    15                      ; carriage return?
                                          GO      [GO     CHRINS          ; yes - strip off the line feed also
                                                   JFCL
003414' 260 17 0 00 005742'                        RTN]
003415' 200 00 0 00 006117'               MOVE    FINSIX                  ; get sixbit command
003416' 350 00 0 17 777776                AOS     -2(P)                   ; set up return + 1
003417' 262 17 0 00 000002                RGET    (2,1)                   ; restore AC's
003421' 263 17 0 00 000000                RTN                             ; return

                                  ; Exit with error or EOF condition

003422' 260 17 0 00 003351'       FINCMX: GO      FIOFF                   ; clear 'file input' switch
003423' 400 00 0 00 000000                SETZ    0,                      ; clear command typed
003424' 262 17 0 00 000002                RGET    (2,1)                   ; restore AC's
003426' 263 17 0 00 000000                RTN                             ; return
```

```

                                      ;#*********************************************************************
                                      ;* FSARG - Input a sixbit argument from a file
                                      ;
                                      ; Return +1 - Error/EOF
                                      ;        +2 - Sixbit word in AC0
                                      ;#*********************************************************************

003427' 261 17 0 00 000001            FSARG:  RPUT    (1,2)                   ; save AC's
003431' 402 00 0 00 006117'                   SETZM   FINSIX                  ; initialize word
003432' 400 01 0 00 000000                    SETZ    1,                      ; byte count
003433' 200 02 0 00 005741'                   MOVE    2,[POINT 6,FINSIX]      ; initial byte pointer

003434' 260 17 0 00 000621'           FSARGO: GO      CHRINS                  ; get a character
003435' 254 00 0 00 003463'                    JRST   FSARGX                  ; exit - error or EOF occurred
003436' 306 00 0 00 000040                    CAIN    " "                     ; space typed?
003437' 254 00 0 00 003453'                   JRST    FSARGC                  ; yes - exit valid command in AC0
003440' 306 00 0 00 000054                    CAIN    ","                     ; comma typed?
003441' 254 00 0 00 003453'                   JRST    FSARGC                  ; yes - exit valid command in AC0
003442' 306 00 0 00 000015                    CAIN    15                      ; carriage return typed?
003443' 254 00 0 00 003453'                   JRST    FSARGC                  ; yes - exit valid command in AC0
003444' 350 00 0 00 000001                    AOS     1                       ; increment byte count
003445' 303 00 0 00 000140                    CAILE   140                     ; lowercase?
003446' 275 00 0 00 000040                    SUBI    40                      ; yes - make into upper case
003447' 275 00 0 00 000040                    SUBI    40                      ; convert to sixbit
003450' 307 01 0 00 000006                    CAIG    1,6                     ; exceed 6 characters?
003451' 136 00 0 00 000002                    IDPB    2                       ; no - save sixbit byte
003452' 254 00 0 00 003434'                   JRST    FSARGO                  ; get next character

                                      ; Exit with command in AC0

003453' 202 00 0 00 030231            FSARGC: MOVEM   $TTCHR                  ; save last character typed
003454' 306 00 0 00 000015                    CAIN    15                      ; carriage return?
                                              GO      [GO     CHRINS          ; yes - strip off the line feed also
                                                        JFCL
003455' 260 17 0 00 005742'                             RTN]
003456' 200 00 0 00 006117'                   MOVE    FINSIX                  ; get sixbit command
003457' 350 00 0 17 777776                    AOS     -2(P)                   ; set up return + 1
003460' 262 17 0 00 000002                    RGET    (2,1)                   ; restore AC's
003462' 263 17 0 00 000000                    RTN                             ; return

                                      ; Exit with error or EOF condition

003463' 260 17 0 00 003351'           FSARGX: GO      FIOFF                   ; clear 'file input' switch
003464' 400 00 0 00 000000                    SETZ    0,                      ; clear command typed
003465' 262 17 0 00 000002                    RGET    (2,1)                   ; restore AC's
003467' 263 17 0 00 000000                    RTN                             ; return
```

```

                                              ;#**********************************************************************
                                              ;* FOARG - Input a octal argument from a file
                                              ;
                                              ; Return +1 - Error/EOF
                                              ;        +2 - Octal data in AC0
                                              ;#**********************************************************************

003470' 261 17 0 00 000001    FOARG:  RPUT    (1,2,3)                 ; save AC's
003473' 400 01 0 00 000000            SETZ    1,                      ; initialize byte count
003474' 403 02 0 00 000003            SETZB   2,3                     ; initialize octal and decimal result
003475' 476 00 0 00 006120'           SETOM   FOFLAG#                 ; set 'octal' flag
003476' 260 17 0 00 000621'   FOARG0: GO      CHRINS                  ; get a character
003477' 254 00 0 00 003544'             JRST  FOARGX                  ; exit - error or EOF occurred
003500' 306 00 0 00 000040            CAIN    " "                     ; space typed?
003501' 254 00 0 00 003530'           JRST    FOARGC                  ; yes - exit valid command in AC0
003502' 306 00 0 00 000054            CAIN    ","                     ; comma typed?
003503' 254 00 0 00 003530'           JRST    FOARGC                  ; yes - exit valid command in AC0
003504' 306 00 0 00 000015            CAIN    15                      ; carriage return typed?
003505' 254 00 0 00 003530'           JRST    FOARGC                  ; yes - exit valid command in AC0
003506' 306 00 0 00 000056            CAIN    "."                     ; decimal point typed?
003507' 402 00 0 00 006120'           SETZM   FOFLAG                  ; yes - clear octal flag
003510' 306 00 0 00 000056            CAIN    "."                     ; decimal point typed?
003511' 254 00 0 00 003476'           JRST    FOARG0                  ; yes - keep looking for delimiter
003512' 350 00 0 00 000001            AOS     1                       ; increment byte count
003513' 303 01 0 00 000014            CAILE   1,^D12                  ; exceed 12 digits?
003514' 254 00 0 00 003544'           JRST    FOARGX                  ; yes - exit - error
003515' 275 00 0 00 000060            SUBI    60                      ; convert to octal 0..n
003516' 331 00 0 00 000000            SKIPL   0                       ; decimal or octal?
003517' 303 00 0 00 000011            CAILE   ^D9                     ; (between 0 and 9)
003520' 254 00 0 00 003544'           JRST    FOARGX                  ; no - exit - error
003521' 221 02 0 00 000012            IMULI   2,^D10                  ; update decimal number
003522' 272 00 0 00 000002            ADDM    2                       ;  ...
003523' 242 03 0 00 000003            LSH     3,3                     ; update octal number
003524' 272 00 0 00 000003            ADDM    3                       ;  ...
003525' 303 00 0 00 000007            CAILE   7                       ; decimal number?
003526' 402 00 0 00 006120'           SETZM   FOFLAG                  ; yes - clear octal flag
003527' 254 00 0 00 003476'           JRST    FOARG0                  ; get next character
```

```

                                        ; Exit with command in AC0

003530' 476 00 0 00 030510      FOARGC: SETOM   TTNBRF                          ; indicate that a number was typed
003531' 202 00 0 00 030231              MOVEM   $TTCHR                          ; save last character typed
003532' 306 00 0 00 000015              CAIN    15                              ; carriage return?
                                        GO      [GO     CHRINS                  ; yes - strip off the line feed also
                                                 JFCL
003533' 260 17 0 00 005742'                      RTN]
003534' 200 00 0 00 000002              MOVE    2                               ; get decimal number
003535' 332 00 0 00 006120'             SKIPE   FOFLAG                          ; octal number?
003536' 200 00 0 00 000003              MOVE    3                               ; yes - get octal number
003537' 350 00 0 17 777775              AOS     -3(P)                           ; set up return + 1
003540' 262 17 0 00 000003              RGET    (3,2,1)                         ; restore AC's
003543' 263 17 0 00 000000              RTN                                     ; return

                                        ; Exit with error or EOF condition

003544' 260 17 0 00 003351'     FOARGX: GO      FIOFF                           ; clear 'file input' switch
003545' 400 00 0 00 000000              SETZ    0,                              ; clear command typed
003546' 262 17 0 00 000003              RGET    (3,2,1)                         ; restore AC's
003551' 263 17 0 00 000000              RTN                                     ; return


                                        ;#*********************************************************************
                                        ;* FEOL - Skip characters until end of line is reached
                                        ;#*********************************************************************

003552' 261 17 0 00 000000      FEOL:   PUT     0                               ; save AC0
003553' 260 17 0 00 000621'     FEOL0:  GO      CHRINS                          ; get a character
003554' 254 00 0 00 003557'             JRST    FEOLX                           ; error/EOF - exit
003555' 302 00 0 00 000012              CAIE    12                              ; LF yet?
003556' 254 00 0 00 003553'             JRST    FEOL0                           ; no - keep looking
003557' 262 17 0 00 000000      FEOLX:  GET     0                               ; restore AC0
003560' 263 17 0 00 000000              RTN                                     ; return

```

```
                        SUBTTL  Scope Looping And Error Reporting Routine

                ;#**********************************************************************
                ;* SCOPE1 - Error reporter and scope loop controller
                ;
                ; This routine is responsible for error reporting and scope looping.
                ; Error looping and print options are controlled by switches.
                ;
                ; Switch Options:
                ; --------------
                ;
                ; DING - 0 No bell on error
                ;        1 Ring bell on error (forced)
                ;
                ; LOOPER - 0 Proceed on error
                ;          1 Loop on first occurence of an error
                ;
                ; ERSTOP - 0 Don't halt on error
                ;          1 Halt on error (exec mode only)
                ;
                ; PALERS - 0 Print error data on first occurance of error only
                ;          1 Print error data for all repetitive errors
                ;
                ; The following two locations should be initialized by the mainline to 0
                ; at the start of each test:    ERFLG   Error/no-error flag
                ;                               ERCNT   A local error counter
                ;                               ERRPC   PC of last error
                ;
                ; The scope/error handler  is called by executing a UUO whose E-field is a
                ; pointer to additional error information. The UUO has a skip and non-skip
                ; return  to  provide the necessary  looping  capabilities.  The  non-skip
                ; return is the  loop  on error return and the skip return is the continue
                ; test  (or  no error)  return.  Pgm  branching is accomplished by placing
                ; appropriate JRSTs in the return locations.
                ;
                ; Example:      CONSZ   DEV,ARG         ; test instruction
                ; -------       SETOM   ERFLG           ; detected an error
                ;               SCOPE   TAG             ; call the routine via UUO
                ;               JRST    ERADR           ; loop return
                ;               JRST    OKADR           ; continue return
                ;
                ;       TAG:    Cntrl Bits,,ADDR        ; variable # of error descriptor
                ;               Cntrl Bits,,ADDR        ;   entries
                ;                ...
```

```

; Each word specifies messages or print routines.  Each  message  printed
; is followed by a CRLF.  The right half of each word is a pointer to the
; start of the  ASCIZ  message or print routine.  The left half furnishes
; message control  information to SCOPE, decoded as follows:
;
;   1B2 (MSG) - Right half of word is address of an ASCIZ message.
;
;   2B2 (CALL) - Right half of word is address of a special print
;                routine.  SCOPE saves AC's 0,1,2,3 so this routine
;                does not have to preserve these.
;
;   3B2 (GOTO) - Right half of word is address where to resume this
;                error descriptor table.
;
;   1B3 (TXNOT) - Print/call entry if TXTINH not set
;
;   1B4 (TXYES) - Print/call entry if TXTINH set
;
;   3B4 (TXALL) - Print/call entry always
;
;   1B5 (LAST) - Right half of word is address of an ASCIZ message.
;
; These bits allows short and normal printout modes, branching, and two
; level module callout priority.
;
; Example: (TXTINH is not set)
;
;        +-+-+-+-+-+-+
;        ERROR: TEST-XXX (Ebus test)
;        SCOPE PC = XXXXXX   TEST PC = XXXXXX   SWITCHES = XXXXXX XXXXXX
;        Error description ...
;        Error description ...
;                ...
;        ------------
;
; Example: (TXTINH is set)
;
;        +-+-+-+-+-+-+
;        ERROR:  TST-XXX SCOPE PC-XXXXXX SW-XXXXXX XXXXXX
;        Error description ...
;        Error description ...
;                ...
;        ------------
;
; NOTE:  If the diagnostic is in debug mode (UDEBUG switch set), no error
;  messages are normally printed.  With MDEBUG switch set, the error flag
;  is set thereby causing each error message to be printed.
;#**********************************************************************
```

```

003561' 261 17 0 00 030114          SCOPE1: PUT     $SVUPC                  ; save the UUO PC
003562' 261 17 0 00 030113                  PUT     $SVUUO                  ; save the actual UUO
003563' 261 17 0 00 000000                  RPUT    (0,1,2)                 ; save some AC's

                                    ; The stack now looks like this:
                                    ;
                                    ;   -5(P) The return addr for SCOPE UUO
                                    ;   -4(P) The address from which the SCOPE UUO was called
                                    ;   -3(P) The actual UUO that was executed "SCOPE"
                                    ;   -2(P) AC0 from before the UUO
                                    ;   -1(P) AC1 from before the UUO
                                    ;    0(P) AC2 from before the UUO

                                    ; Get switches

003566' 260 17 0 00 001571'                 GO      SWITT                   ; get switches
003567' 202 00 0 00 006133'                 MOVEM   SCOSW#                  ; save switches

                                    ; Handle SPEAR error report

003570' 332 00 0 00 000015                  SKIPE   ERFLG                   ; error occurred?
003571' 260 17 0 00 000000*                 GO      SPREP2                  ; yes - report it

                                    ; Handle debug printout / error handling

003572' 332 00 0 00 000000*                 SKIPE   UDEBUG                  ; no-port debug mode?
003573' 400 15 0 00 000000                  SETZ    ERFLG,                  ; yes - clear error flag
003574' 332 00 0 00 000000*                 SKIPE   MDEBUG                  ; error message debug mode?
003575' 474 15 0 00 000000                  SETO    ERFLG,                  ; yes - set error flag

                                    ; Update scope PC

003576' 550 00 0 17 777774                  HRRZ    -4(P)                   ; get PC of the scope call
003577' 202 00 0 00 006132'                 MOVEM   SCOPC#                  ; save it (scope PC)

                                    ; Error occurred?

003600' 322 15 0 00 003635'                 JUMPE   ERFLG,SCOP4             ; an error occurred? - no check looping
```

```

                                    ; Error did occur - update error PC / print error message

003601' 316 00 0 00 030052                  CAMN    ERRPC                   ; error has already occurred here?
003602' 254 00 0 00 003610'                 JRST    SCOPO                   ; yes - go check if print
003603' 200 00 0 00 006132'                 MOVE    SCOPC                   ; get scope PC
003604' 202 00 0 00 030052                  MOVEM   ERRPC                   ; save as error PC
003605' 200 00 0 00 000516*                 MOVE    TSTSUB                  ; get current offset
003606' 202 00 0 00 004062'                 MOVEM   SCOOFF                  ; save as error offset
003607' 254 00 0 00 003616'                 JRST    SCOP2                   ; go print error

                                    ; Must also check if this is a multiple error from a execute table.  Check
                                    ; TSTSUB against SCOOFF - if different then it is not a repeat error.

003610' 200 00 0 00 004062'         SCOPO:  MOVE    SCOOFF                  ; last offset presented to SCOPE
003611' 312 00 0 00 003605*                 CAME    TSTSUB                  ; same as current offset?
003612' 254 00 0 00 003616'                 JRST    SCOP2                   ; yes - go check if print

                                    ; This is a repeat error here, check if print all errors

003613' 200 00 0 00 006133'         SCOP1:  MOVE    SCOSW                   ; get switches
003614' 607 00 0 00 001000                  TLNN    PALERS                  ; print all errors?
003615' 254 00 0 00 003626'                 JRST    SCOP3                   ; no - go ring bell/error halt

                                    ; Print error message

003616' 200 00 0 00 006133'         SCOP2:  MOVE    SCOSW                   ; get switches
003617' 602 00 0 00 100000                  TRNE    INHMSG                  ; inhibit error messages?
003620' 254 00 0 00 003626'                 JRST    SCOP3                   ; yes - skip this
003621' 260 17 0 00 003723'                 GO      ERHDR                   ; print the standard error heading
003622' 550 01 0 17 777775                  HRRZ    1,-3(P)                 ; get the e-field of the UUO
003623' 332 00 0 00 000001                  SKIPE   1                       ; table address given?
003624' 260 17 0 00 003776'                 GO      ERPNT                   ; yes - go handle error table
003625' 037 00 0 00 005745'                 TMSGCD  <------------->

                                    ; Handle bell and error halt functions.

003626' 200 00 0 00 006133'         SCOP3:  MOVE    SCOSW                   ; get switches
003627' 603 00 0 00 010000                  TLNE    DING                    ; ding set?
003630' 037 01 0 00 000007                  PBELL                           ; yes - do it
003631' 200 00 0 00 006133'                 MOVE    SCOSW                   ; get switches
003632' 607 00 0 00 002000                  TLNN    ERSTOP                  ; halt on error?
003633' 254 00 0 00 003635'                 JRST    SCOP4                   ; no - continue
003634' 476 00 0 00 000000*                 SETOM   ALTF                    ; yes - set abort flag
```

```

                                        ; Error reporting (if any) is done.  Now check loop on error switch.

003635' 200 00 0 00 006133'             SCOP4:  MOVE    SCOSW                   ; get switches
003636' 607 00 0 00 004000                      TLNN    LOOPER                  ; loop on error?
003637' 254 00 0 00 003666'                     JRST    SCOPXX                  ; no - exit +2/3
003640' 326 15 0 00 003646'                     JUMPN   ERFLG,SCOPX1            ; yes - error - yes - exit +1 (loop)

                                        ; Error looping is enabled, but error flag was not set.  First check if we
                                        ; are at the same place as the last error - if not, exit +3 because it is
                                        ; not yet time to loop.  If yes, exit +1, loop on error.

003641' 200 00 0 00 006132'                     MOVE    SCOPC                   ; get scope PC and see if this is
003642' 315 00 0 00 030052                      CAMGE   ERRPC                   ;    earlier than the error PC?
003643' 254 00 0 00 003666'                     JRST    SCOPXX                  ; earlier - exit +3
003644' 336 00 0 00 030052                      SKIPN   ERRPC                   ; no error yet at all?
003645' 254 00 0 00 003666'                     JRST    SCOPXX                  ; no - exit +3

                                        ; Exit +1 (loop return)

003646' 037 07 0 00 000003              SCOPX1: TTALTM                          ; see if opr wants to abort
003647' 334 00 0 00 000000                      SKIPA                           ; no - continue
003650' 254 00 0 00 003673'                     JRST    SCOPX2                  ; yes - altmode exit
003651' 332 00 0 00 003634*                     SKIPE   ALTF                    ; 'altmode typed' flag set?
003652' 254 00 0 00 003673'                     JRST    SCOPX2                  ; yes - altmode exit
003653' 336 00 0 00 006126'                     SKIPN   LCHAR                   ; last character in here already?
003654' 202 00 0 00 006126'                     MOVEM   LCHAR                   ; no - save 'last char typed'
003655' 260 17 0 00 000474'                     GO      LSTAMP                  ; check for S typed
003656' 200 00 0 00 006132'                     MOVE    SCOPC                   ; get scope PC
003657' 202 00 0 00 030052                      MOVEM   ERRPC                   ; update error PC
003660' 262 17 0 00 000002                      RGET    (2,1,0,$SVUUO,$SVUPC)   ; restore things
003665' 263 17 0 00 000000                      RTN                             ; exit

                                        ; Exit +2/3 (altmode return / normal return)

003666' 037 07 0 00 000003              SCOPXX: TTALTM                          ; see if opr wants to abort
003667' 334 00 0 00 000000                      SKIPA                           ; no - continue
003670' 254 00 0 00 003673'                     JRST    SCOPX2                  ; yes - altmode exit
003671' 336 00 0 00 003651*                     SKIPN   ALTF                    ; 'altmode typed' flag set?
003672' 254 00 0 00 003703'                     JRST    SCOPX3                  ; no - exit +3 (normal return)

                                        ; Altmode return

003673' 262 17 0 00 000002              SCOPX2: RGET    (2,1,0,$SVUUO,$SVUPC)   ; restore things
003700' 350 00 0 17 000000                      AOS     (P)                     ; set up return +1
003701' 476 00 0 00 003671*                     SETOM   ALTF                    ; set 'altmode typed' flag
003702' 263 17 0 00 000000                      RTN                             ; exit
```

```
.COMM5
.COMM6
.CTBL
.CTYP
.DARG
.DARG0
.DARG1
.DARGA
.DARX1
.DARX2
.EXIT
.FHSLF
.ISWT
.ISWT2
.ISWT6
.ISWT7
.ISWT8
.ISWTB
.ISWTX
.ISWTY
.ISWTZ
.LSWIT
.LSWTN
.OARG
.OARG0
.OARG1
.OARGA
.OARX1
.OARX2
.PNTOC
.RESET
.RSWIT
.SARG
.SARG0
.SARG1
.SARGA
.SARX1
.SARX2
.SLEFT
.SLF1
.SLF2
.SLF3
.SLF4
.SLF5
.SLF6
.SLFX
.SRG2
.SRG3
.SRG4
.SRG5
.SRG6
.SRG6A
.SRG6B
```

```


                                        ; Normal return - also check for S typed

003703' 336 00 0 00 006126'     SCOPX3: SKIPN   LCHAR                           ; last character in here already?
003704' 202 00 0 00 006126'             MOVEM   LCHAR                           ; no - save 'last char typed'
003705' 260 17 0 00 000423'             GO      PSTAMP                          ; check for S typed
003706' 200 00 0 00 006133'             MOVE    SCOSW                           ; get switches
003707' 602 00 0 00 010000              TRNE    RUNALL                          ; run all test segments?
003710' 400 15 0 00 000000              SETZ    ERFLG,                          ; yes - clear error flag
003711' 602 00 0 00 001000              TRNE    DEBUG2                          ; error message debug?
003712' 400 15 0 00 000000              SETZ    ERFLG,                          ; yes - clear error flag
003713' 262 17 0 00 000002              RGET    (2,1,0,$SVUUO,$SVUPC)           ; restore things
003720' 350 00 0 17 000000              AOS     (P)                             ; set up return +3
003721' 350 00 0 17 000000              AOS     (P)                             ; set up return +3
003722' 263 17 0 00 000000              RTN                                     ; exit
```

```



                                        ;#**********************************************************************
                                        ; ERHDR - Print the standard header messages
                                        ;#**********************************************************************

003723' 037 01 0 00 000026              ERHDR:  PFORCE                                  ; handle Control-O
                                                TMSGC   <+-+-+-+-+-+-+
003724' 037 00 0 00 005751'             ? >
003725' 260 17 0 00 004045'                     GO      PNTDEV                          ; print device type
003726' 037 00 0 00 005755'                     TMSG    < Error:  TEST >
003727' 260 17 0 00 000000*                     GO      TSTPNT                          ; print test name
003730' 200 00 0 00 006133'                     MOVE    SCOSW                           ; get switches
003731' 603 00 0 00 000200                      TLNE    TXTINH                          ; short printout desired?
003732' 254 00 0 00 003757'                     JRST    ERSHRT                          ; yes - go handle

                                        ; Regular format error header

003733' 037 00 0 00 005760'             ERNRML: TMSGC   <Test PC = >
003734' 200 00 0 00 000000*                     MOVE    TSTPC                           ; get the address of start of test
003735' 274 00 0 00 000000*                     SUB     TSTOFF                          ; normalize to relocatable address
003736' 037 16 0 00 000003                      PNTOCS                                  ; print it
003737' 037 00 0 00 005763'                     TMSG    <'  Scope PC = >
003740' 550 00 0 17 777772                      HRRZ    -6(P)                           ; get adr where SCOPE was called
003741' 274 00 0 00 003735*                     SUB     TSTOFF                          ; normalize to relocatable address
003742' 037 16 0 00 000003                      PNTOCS                                  ; print it
003743' 037 00 0 00 005766'                     TMSG    <'>
003744' 336 00 0 00 000461*                     SKIPN   TSTREP                          ; pass 1?
003745' 254 00 0 00 003753'                     JRST    ERNRMO                          ; yes - don't print pass #
003746' 037 00 0 00 005767'                     TMSG    <  Pass >
003747' 200 00 0 00 003744*                     MOVE    TSTREP                          ; get pass count
003750' 350 00 0 00 000000                      AOS
003751' 037 15 0 00 000000                      PNTDEC                                  ; print it
003752' 037 00 0 00 004077'                     TMSG    <.>
003753' 037 00 0 00 005771'             ERNRMO: TMSG    <  Switches = >
003754' 200 00 0 00 006133'                     MOVE    SCOSW                           ; get switches
003755' 037 13 0 00 000000                      PNTHW                                   ; print
003756' 263 17 0 00 000000                      RTN                                     ; exit
```

```

                                        ; Short format error header

003757' 037 00 0 00 005774'     ERSHRT: TMSG    <  Scope PC >
003760' 550 00 0 17 777772              HRRZ    -6(P)                   ; pick up error PC
003761' 274 00 0 00 003741*             SUB     TSTOFF                  ; normalize to relocatable address
003762' 037 16 0 00 000003              PNTOCS                          ; print the error PC
003763' 037 00 0 00 005766'             TMSG    <'>
003764' 336 00 0 00 003747*             SKIPN   TSTREP                  ; pass 1?
003765' 254 00 0 00 003772'             JRST    ERSHRO                  ; yes - don't print pass #
003766' 037 00 0 00 005767'             TMSG    <  Pass >
003767' 200 00 0 00 003764*             MOVE    TSTREP                  ; get pass count
003770' 350 00 0 00 000000              AOS
003771' 037 15 0 00 000000              PNTDEC                          ; print it
003772' 037 00 0 00 005777'     ERSHRO: TMSG    <  SW >
003773' 200 00 0 00 006133'             MOVE    SCOSW                   ; get switches
003774' 037 13 0 00 000000              PNTHW                           ; print
003775' 263 17 0 00 000000              RTN                             ; exit


                                ;#*********************************************************************
                                ; ERPNT - Print error description based on error descriptor table
                                ;#*********************************************************************

003776' 202 01 0 00 006116'     ERPNT:  MOVEM   1,ERDES#                ; save addr of error descriptor table
003777' 336 01 1 00 006116'     ERPNO:  SKIPN   1,@ERDES                ; get entry - zero?
004000' 263 17 0 00 000000              RTN                             ; yes - exit

004001' 350 00 0 00 006116'             AOS     ERDES                   ; update to point at next one
004002' 135 02 0 00 006001'             LDB     2,[POINT 3,1,2]         ; get entry type
                                        GO      @[ERPNE                 ; yes - dispatch
                                                  ERPNM                 ; handle message
                                                  ERPNC                 ; handle call
                                                  ERPNG                 ; handle goto
                                                  ERPNE                 ; out of range
                                                  ERPNE                 ; out of range
                                                  ERPNE                 ; out of range
004003' 260 17 1 02 006002'                       ERPNE](2)             ; out of range
004004' 607 01 0 00 010000              TLNN    1,(LAST)                ; last entry?
004005' 254 00 0 00 003777'             JRST    ERPNO                   ; no - get next entry
004006' 263 17 0 00 000000              RTN                             ; yes - return

                                ; Error (entry type out of range)

004007' 037 00 0 00 006012'     ERPNE:  TMSGC   <? Error descriptor entry type out of range>
004010' 263 17 0 00 000000              RTN                             ; return
```

```

                                        ; Handle message

004011' 200 00 0 00 006133'     ERPNM:  MOVE    SCOSW                           ; get switches
004012' 603 00 0 00 000200              TLNE    TXTINH                          ; text inhibit set?
                                        PJRST   [TLNN   1,(TXYES)               ; yes - print it?
                                                 JRST   .+1                     ; no - exit
                                                 PCRL                           ; yes - print it
                                                 PNTMSG @1                      ;   and return
004013' 254 00 0 00 006023'                      RTN]
004014' 607 00 0 00 000200              TLNN    TXTINH                          ; text inhibit set?
                                        PJRST   [TLNN   1,(TXNOT)               ; no - print it?
                                                 RTN                            ; no - return
                                                 PCRL                           ; yes - print it
                                                 PNTMSG @1                      ;   and return
004015' 254 00 0 00 006030'                      RTN]
004016' 263 17 0 00 000000              RTN                                     ; return

                                ; Handle goto

004017' 552 02 0 00 006116'     ERPNG:  HRRZM   2,ERDES                         ; update table address
004020' 263 17 0 00 000000              RTN                                     ; exit

                                ; Handle call

004021' 200 00 0 00 006133'     ERPNC:  MOVE    SCOSW                           ; get switches
004022' 603 00 0 00 000200              TLNE    TXTINH                          ; text inhibit set?
                                        JRST    [TLNN   1,(TXYES)               ; yes - call routine?
                                                 JRST   .+1                     ; no - exit
004023' 254 00 0 00 006035'                      JRST   ERPNCG]                 ; yes - call routine
004024' 607 00 0 00 000200              TLNN    TXTINH                          ; text inhibit set?
                                        PJRST   [TLNN   1,(TXNOT)               ; no - call routine?
                                                 RTN                            ; no - return
004025' 254 00 0 00 006040'                      JRST   ERPNCG]                 ; yes - call routine
004026' 263 17 0 00 000000              RTN                                     ; return
004027' 261 17 0 00 000000      ERPNCG: RPUT    (0,1,2,3,4,5)                   ; save some AC's
004035' 260 17 1 00 000001              GO      @1                              ; call the routine
004036' 262 17 0 00 000005              RGET    (5,4,3,2,1,0)                   ; restore AC's
004044' 263 17 0 00 000000              RTN                                     ; exit
```

```


                                          ;#********************************************************************
                                          ; PNTDEV - Print device name
                                          ;#********************************************************************

 004045' 135 01 0 00 006043'              PNTDEV: LDB     1,[POINT 3,UUT,9]       ; get channel number
                                                  XCT     [TMSG   <RH0>
                                                           TMSG   <RH1>
                                                           TMSG   <RH2>
                                                           TMSG   <RH3>
                                                           TMSG   <RH4>
                                                           GO     PNTDNI
                                                           TMSG   <RH6>
 004046' 256 00 0 01 006052'                               GO     PNTDCI](1)
 004047' 263 17 0 00 000000                       RTN

 004050' 332 00 0 00 000000*              PNTDNI: SKIPE   NISEL                   ; NI exists?
 004051' 037 00 0 00 006062'                      TMSG    <NI>                    ; yes - print such
 004052' 336 00 0 00 004050*                      SKIPN   NISEL                   ; NI exists?
 004053' 037 00 0 00 006063'                      TMSG    <RH5>                   ; no - print such
 004054' 263 17 0 00 000000                       RTN

 004055' 332 00 0 00 000000*              PNTDCI: SKIPE   CISEL                   ; CI exists?
 004056' 037 00 0 00 006064'                      TMSG    <CI>                    ; yes - print such
 004057' 336 00 0 00 004055*                      SKIPN   CISEL                   ; CI exists?
 004060' 037 00 0 00 006065'                      TMSG    <RH7>                   ; no - print such
 004061' 263 17 0 00 000000                       RTN

                                          ; Miscellaneous (used for tests that have an execute table)

 004062' 000000  000000                   SCOOFF: 0                               ; error offset



                                          ;#********************************************************************
                                          ; End Stmt for DFPTAU.MAC module
                                          ;#********************************************************************

                                                  XLIST

NO ERRORS DETECTED

PROGRAM BREAK IS 006142
CPU TIME USED 00:12.646

101P CORE USED
```

```
DFPTAU  MACRO %53B(1242) 10:16  1-Jul-85 Page S-1
DFPTAU  MAC      16-May-85 17:43         SYMBOL TABLE                                                SEQ 2151

ABUFF            006112'      DECERR           003214'      FREAD   037300  000004        PATCN8          001334'
ALTCHH           000031' ent  DECLIS           003220'      FSARG           003427'       PATFLG          006130'
ALTCHK  007000   000000       DECYN            003202' ent  FSARG0          003434'       PATPNT          001402' ent
ALTCHX           000042'      DING             010000  spd  FSARGC          003453'       PATRA1          001325'
ALTF             005704' ext  DROPDV  037240   000002       FSARGX          003463'       PATRAN          001322'
ALTMGO           030063       ERCAL   320740   000000  int  FSELEC  037240  000004        PBELL   037040  000007
ARGFLG           003277' int  ERDES            006116'      FSKIPC          000742'       PCRL    037000  030242
ARGUM            003276' int  ERFLG            000015       GENFLG          006122'       PCRL2F  037040  030243
BUFCOM           000775' ent  ERHDR            003723'      GENREP          006123'       PCRLF   037040  030242
BUFCR1           001122'      ERNRM0           003753'      GENTYP          006124'       PDEC    006000  000000
BUFCRW           001104'      ERNRML           003733'      GET     262740  000000        PDEC0           000172'
BUFF             001331' ext  ERPN0            003777'      GETLOG          002732' ext   PDECF0          000220'
BUFG0            001037'      ERPNC            004021'      GETTAB  047000  000041        PDECFS          000230'
BUFG1            001046'      ERPNCG           004027'      GO      260740  000000        PDECSS          000202'
BUFG2            001053'      ERPNE            004007'      HALTF   104000  000170  int   PFORCE  037040  000026
BUFGEN           001006' ent  ERPNG            004017'      INDLIS          000512' ext   PJRST   254000  000000
BUFGX            001061'      ERPNM            004011'      INHMSG          100000  spd   PNTCHF  037500  000001
BUFGXY           001073'      ERPNT            003776'      INITM           006125'       PNTCI   037000  000000
BUFGXZ           001075'      ERRPC            030052       ITRCH1          030104        PNTCIF  037040  000000
CADDR            000671' ext  ERSHR0           003772'      JEN     254500  000000        PNTDCF  037640  000001
CFIBF   104000   000100       ERSHRT           003757'      KLFLG           030041        PNTDCI          004055'
CHKAR0           003311'      ERSTOP           002000  spd  LASAR0          003335'       PNTDEC  037640  000000
CHKARA           003320'      EXT10            000572'      LASAR1          003341'       PNTDEV          004045' ent
CHKARG           003300' ent  EXTX             000574'      LASARA          003345'       PNTDNI          004050'
CHKARX           003317'      FEOL             003552'      LASARG          003324' ent   PNTDS   037700  000000
CHKARY           003315'      FEOL0            003553'      LASARX          003343'       PNTDSF  037700  000001
CHRIN            000614'      FEOLX            003557'      LAST    010000  000000  spd   PNTHW   037540  000000
CHRINS           000621'      FFAIL            000645'      LCHAR           006126' int   PNTHWF  037540  000001
CHRINX           000633'      FFF              002745' ext  LOGPNT          002733' ext   PNTMSF  037040  000000
CISEL            004057' ext  FGETA            000726'      LOOPER          004000  spd   PNTMSG  037000  000000
CLRBFI  051440   000000       FGETA0           000727'      LSTAM0          000510'       PNTOCF  037740  000003
CLRBUF           000045' ent  FGETAE           000740'      LSTAMP          000474'       PNTOCS  037700  000003
CLS10            000555'      FGETD            000750'      LSTAMX          000525'       PNTSIX  037000  000002
CLSX             000557'      FGETD0           000754'      MAXPAT          000127  spd   PNTSXF  037040  000002
CLZFF   104000   000034  int  FGETD1           000771'      MAXTIM          000420' int   PORSEL          000000  ext
CONSW            030046       FGETDE           000772'      MDEBUG          003574' ext   PORTCI          000000  ext
CONSWW           006113'      FGETW            000661' ent  MONTI0          000276'       PORTNI          000000  ext
CONVS0           000006'      FGETWE           000722'      MONTI1          000301'       PPDEC           000163' ent
CONVS1           000022'      FILLFP           000115  spd  MONTIM          000262' ent   PPDECF          000211' ent
CONVSE           000017'      FILLVP           000127  spd  MONTYP          030516        PSDN            000105' ent
CONVSP           000027'      FINCM0           003373'      NISEL           004052' ext   PSDNF           000113'
CONVSS           006114'      FINCMC           003412'      OLDRAN          006127'       PSTAM0          000437'
CONVSX           000000' ent  FINCMD           003366' ent  OPRSEL          000010  spd   PSTAMP          000423' ent
CORECT           002746'      FINCMX           003422'      P               000017        PSTAMX          000465'
CURTIM           000417' int  FINECH           000627' ext  PALERS          001000  spd   PTIME           000311' ent
CWORDL           004332' ext  FINIT            000576' ent  PARSER          002717' ent   PTIMEF          000317'
CWORDN           006115'      FINPUT           003363' ext  PASCNT          030047        PUT     261740  000000
CWORDR           004335' ext  FINSIX           006117'      PAT             000014        RADLSC          030264
CYCL60           030505       FIOFF            003351' ent  PATADR          001331'       RANDBS          030022
DATM0            001140'      FOARG            003470' ent  PATBUF          001172' ent   RDCSR           002724' ext
DATM1            001146'      FOARG0           003476'      PATCN2          004372' spd   RESET   104000  000147  int
DATMES           001132'      FOARGC           003530'      PATCN3          004373' spd   RTN     263740  000000
DATMX            001165'      FOARGX           003544'      PATCN4          004374' spd   RUNALL          010000  spd
DEBTIM           000421' int  FOFLAG           006120'      PATCN6          001366'       RUNTM0          000335'
DEBUG2           001000  spd  FORPNT           006121' int  PATCN7          001351'       RUNTM1          000337'
```

```
DFPTAU  MACRO %53B(1242) 10:16  1-Jul-85 Page S-2
DFPTAU  MAC        16-May-85 17:43           SYMBOL TABLE                                                   SEQ 2152

RUNTM2          000366'         TXYES   020000  000000  spd     .OARX2          003161'
RUNTME          000325' ent     UDEBUG          003572' ext     .PNTOC          000121' ent
RUNTMX          000410'         UNAME           000657' int     .RESET          000534' ent
SAVSWW          006131'         USER            030037          .RSWIT          002545' int
SCOOFF          004062' int     USTIM           000422' int     .SARG           003056' ent
SCOP0           003610'         UUT             006043' ext     .SARG0          003071'
SCOP1           003613'         VARPAT          006135'         .SARG1          003073'
SCOP2           003616'         VPATRN          000116  spd     .SARGA          003116'
SCOP3           003626'         WRDERR          006136' int     .SARX1          003110'
SCOP4           003635'         XX              006137'         .SARX2          003107'
SCOPC           006132'         XXT             006140'         .SLEFT          002254'
SCOPE1          003561' ent     XYZ             006141'         .SLF1           002263'
SCOPX1          003646'         ZU              000000'         .SLF2           002274'
SCOPX2          003673'         $CHRIN          030232          .SLF3           002300'
SCOPX3          003703'         $SVUPC          030114          .SLF4           002301'
SCOPXX          003666'         $SVUU0          030113          .SLF5           002306'
SCOSW           006133' int     $TTCHR          030231          .SLF6           002341'
SPREP2          003571' ext     $TWCNT          030225          .SLFX           002345'
STCLOK          000237' ent     $TYPNB          030233          .SRG2           002364'
STCU            000251'         .CGOT           003005' int     .SRG3           002375'
STCX            000255'         .CLOSE          000543' ent     .SRG4           002400'
SWCHPT          002567' ent     .COMM           002761' ent     .SRG5           002401'
SWCOM           002576' ent     .COMM0          002766'         .SRG6           002410'
SWCOM0          002615'         .COMM1          002776'         .SRG6A          002436'
SWCOMP          002666'         .COMM2          003012'         .SRG6B          002442'
SWCOMT          002635'         .COMM3          003024'         .SRG7           002452'
SWCOMX          002631'         .COMM4          003034'         .SRGX           002456'
SWITCH  037400  000002          .COMM5          003042'         .SRIGT          002355'
SWITT           001571' ent     .COMM6          003047'         .SWCH0          001547'
SWPSWW          006134'         .CTBL           003010'         .SWCH1          001557'
SWRGT           001576' int     .CTYP           003006'         .SWCHP          001532' ent
TAKFIL          003355' ext     .DARG           003223' ent     .SWIN           002466'
TIME    104000  000014  int     .DARG0          003236'         .SWIN0          002474'
TMRST1          000260'         .DARG1          003240'         .SWPNT          001637'
TMRUN1          000415'         .DARGA          003264'         .SWTXT          001645'
TPAT            004405' ext     .DARX1          003256'
TSTCLS          000511' ext     .DARX2          003255'
TSTINH          000000  ext     .EXIT           000561' ent
TSTNUM          000514' ext     .FHSLF          400000  sin
TSTOFF          003761' ext     .ISWT           001577' ent
TSTPC           003734' ext     .ISWT2          001602'
TSTPNT          003727' ext     .ISWT6          001613'
TSTREP          003767' ext     .ISWT7          001614'
TSTSUB          003611' ext     .ISWT8          001616'
TTALTM  037340  000003          .ISWTB          001620'
TTICNV  037240  000003          .ISWTX          001621'
TTIOCT  037140  000003          .ISWTY          001631'
TTNBRF          030510          .ISWTZ          001634'
TTSIXB  037400  000003          .LSWIT          002511' int
TTYINP  037440  000003          .LSWTN          002527'
TTYY0           000065'         .OARG           003127' ent
TTYY1           000067'         .OARG0          003142'
TTYYES          000061' ent     .OARG1          003144'
TXNOT   040000  000000  spd     .OARGA          003170'
TXTINH          000200  spd     .OARX1          003162'
```

```
ABUFF      1852#      1852
ALTCHH       12        118#
ALTCHX      121        124        128#
ALTF         53#      1965       2052       2167       2255       2299       2689       2713       2728       2735
ALTMGO      130       1965       1968       2017       2052       2055       2105       2167       2170       2220       2253       2258       2285       2297
           2302       2326
ARGFLG       41       1972       1997       2000       2059       2085       2088       2174       2200       2203       2241#      2253       2297
ARGUM        41       1993       2076       2136       2191       2240#
BUFCOM       30        865#
BUFCR1      976#       984
BUFCRW      872        965#
BUFF         50#       899        968        970       1013       1163
BUFG0       908#       918
BUFG1       907        926#
BUFG2       934#       944
BUFGEN       30        874#
BUFGX       919        948#       986
BUFGXY      951        953        958#
BUFGXZ      949        960#
CADDR        50#       763        768
CHKAR0     2265#      2277
CHKARA     2257       2273       2285#
CHKARG       40       2252#
CHKARX     2254       2264       2279       2284#
CHKARY     2268       2280       2282#
CHRIN       699#       701        769        804        816        833
CHRINS      709#      2359       2379       2406       2426       2452       2484       2507
CHRINX      711        720#
CISEL        52#      2896       2898
CLRBUF       11        137#      1513       1579       1621       1642
CLS10       627        632#
CLSX        631        634#
CONSW      1373       1430       1713       1752
CONSWW     1399#      1399       1419       1428       1481       1488       1503       1525       1527       1529       1531       1544       1551       1569
           1597       1599       1601       1603
CONVS0       83#        91
CONVS1       84        100#
CONVSE       87         95#
CONVSP       81         83        105#
CONVSS       79#        79         80
CONVSX       12         78#
CORECT     1853       1857#
CURTIM       17        451        510#
CWORDL       50#       764        775        778        782
CWORDN     1014#      1014
CWORDR       50#       765        785        788        791
CYCL60      374        376
DATM0       995        996#
DATM1      1003       1008#
DATMES      913        939        979        993#
DATMX      1001       1004       1026#
DEBTIM       17        448        512#
DEBUG2     2746
```

```
DECERR    2140   2144#
DECLIS    2137   2147#
DECYN       40   2135#
DING      2684
ERDES     2813#  2813   2814   2817   2855
ERFLG     2628   2634   2636   2645   2696   2745   2747
ERHDR     2675   2757#
ERNRMO    2779   2785#
ERNRML    2769#
ERPNO     2814#  2828
ERPNC     2821   2860#
ERPNCG    2864   2868   2870#
ERPNE     2819   2823   2824   2825   2826   2833#
ERPNG     2822   2855#
ERPNM     2820   2838#
ERPNT     2678   2813#
ERRPC     2649   2652   2703   2705   2719
ERSHRO    2798   2803#
ERSHRT    2765   2792#
ERSTOP    2687
EXT10      654    661#
EXTX       659    664#
FEOL      2263   2304   2506#
FEOLO     2507#  2510
FEOLX     2508   2511#
FFAIL      680    736#
FFF         57#  1835   1847   1855
FGETA      766    803#
FGETAO     804#   809
FGETAE     805    811    813#   817
FGETD      773    776    779    786    789    810    830#
FGETDO     833#   845
FGETD1     836    838    846#
FGETDE     834    841    847#
FGETW       25    762#
FGETWE     767    770    774    777    780    787    790    793#
FILLFP     881   1116#
FILLVP     883   1144#
FINCMO    2359#  2373
FINCMC    2362   2364   2366   2377#
FINCMD      25   2354#
FINCMX    2360   2389#
FINECH      52#   715
FINIT       25    676#
FINPUT      52#   727   1977   2064   2179   2262   2303   2333   2343
FINSIX    2355#  2355   2357   2382   2402   2404   2429
FIOFF       40   2333#  2389   2436   2496
FOARG       25   2065   2180   2448#
FOARGO    2452#  2463   2477
FOARGC    2455   2457   2459   2481#
FOARGX    2453   2466   2470   2496#
FOFLAG    2451#  2451   2461   2476   2488
FORPNT      17    398#   398    405    469    471    474    476    482    484    486    488    494    496
```

```
                498       500
FSARG          1978      2401#
FSARG0         2406#     2420
FSARGC         2409      2411      2413      2424#
FSARGX         2407      2436#
FSKIPC          807       816#      821
GENFLG          866#      866       874       888       909       911       935       937       948
GENREP          867#      867       869       952       994      1000
GENTYP          875#      875
GETLOG           57#     1844
INDLIS           51#      535       575
INHMSG         2673
INITM           343#      343       382#      382       383#      383       384#      384
ITRCH1         1867
KLFLG          1846
LASAR0         2306#     2317
LASAR1         2309      2320      2322#
LASARA         2301      2326#
LASARG           40      2006      2094      2209      2296#
LASARX         2298      2305      2313      2319      2324#
LAST           2827
LCHAR            17       521#      521       554       563#      563       587      2715      2716      2740      2741
LOGPNT           57#     1845
LOOPER         2694
LSTAM0          565       567       573#
LSTAMP          562#     2717
LSTAMX          569       587#
MAXPAT         1148#
MAXTIM           17       511#
MDEBUG           50#     2635
MONTI0          369       379#
MONTI1          378       382#
MONTIM           16       344       367#      439
MONTYP          141       143       368       602       604       626       653
NISEL            52#     2890      2892
OLDRAN         1153#     1153      1158
OPRSEL          547
P               102       172       185       194       209       210       691       702       713       714       792       812       846       959
               1622      1623      1624      1633      1643      1644      1735      1905      1941      1946      1947      2004      2008      2012
               2013      2027      2092      2096      2100      2101      2115      2142      2143      2207      2211      2215      2216      2228
               2272      2284      2324      2325      2383      2430      2490      2640      2676      2734      2749      2750      2774      2793
PALERS         2667
PARSER           40      1834#
PASCNT          548
PAT             876       878       882       891       906       908       926       960       966
PATADR         1130      1163#
PATBUF           30       908       926      1038#     1116      1122      1141      1144      1148
PATCN2         1134#     1138
PATCN3         1135#     1139
PATCN4         1136#     1140
PATCN6         1140      1197#
PATCN7         1139      1183#
PATCN8         1138      1169#
```

```
PATFLG     884#    884     891
PATPNT      30    1212#
PATRA1    1129    1156#
PATRAN    1129    1152#
PDECO      249#    255
PDECFO     285#    291
PDECFS     284     291#
PDECSS     248     255#
PNTDCI    2887    2896#
PNTDEV      11     532    2760    2879#
PNTDNI    2885    2890#
PORSEL      52#
PORTCI      52#
PORTNI      52#
PPDEC       12     245#
PPDECF      12     281#
PSDN        12     183#
PSDNF      192#
PSTAMO     523     525     531#
PSTAMP      16     520#   2742
PSTAMX     527     554#
PTIME       16     396#
PTIMEF     403#    544     584
RADLSC     249     251     285     287
RANDBS    1156
RDCSR       57#   1838
RUNALL    2744
RUNTMO     436     439#
RUNTM1     438     444#
RUNTM2     465     481#
RUNTME      16     399     406     431#
RUNTMX     503#
SAVSWW    1398#   1398    1432
SCOOFF      46    2654    2660    2904#
SCOPO     2650    2660#
SCOP1     2666#
SCOP2     2655    2662    2672#
SCOP3     2668    2674    2683#
SCOP4     2645    2688    2693#
SCOPC     2641#   2641    2651    2702    2718
SCOPE1      45    2608#
SCOPX1    2696    2710#
SCOPX2    2712    2714    2727    2733#
SCOPX3    2729    2740#
SCOPXX    2695    2704    2706    2725#
SCOSW       46    2624#   2624    2666    2672    2683    2686    2693    2743    2763    2786    2804    2838    2860
SPREP2      50#   2629
STCLOK      16     334#
STCU       336     343#
STCX       342     347#
SWCMPT      35    1711#
SWCOM       35    1723#
SWCOMO    1728    1750#
```

```
SWCOMP      1724     1795#
SWCOMT      1725     1767#
SWCOMX      1746     1762#
SWITT         35      546     1312     1315     1373#    1397     1418     1431     1750     1756     2623
SWPSWW      1403#    1403     1434#    1434
SWRGT         36     1376     1379#    1429     1753
TAKFIL        52#     721      724     2337     2340
TMRST1       341      346      350#     445
TMRUN1       437      440      444      508#
TPAT          52#    1142
TSTCLS        51#     534      574
TSTINH        53#
TSTNUM        51#     537      577
TSTOFF        51#    2771     2775     2794
TSTPC         51#    2770
TSTPNT        51#    2762
TSTREP        51#     549     2778     2781     2797     2800
TSTSUB        51#     539      579     2653     2661
TTNBRF      2077     2192     2274     2314     2481
TTYY0        159#     163      170
TTYY1        161#     164
TTYYES        11      157#    1446
TXNOT       2846     2866
TXTINH      2764     2839     2845     2861     2865
TXYES       2840     2862
UDEBUG        50#    2633
UNAME         26      677      679      684      687      738      741      747#
USER         137      335      435      600      622      649     1374     1404     1413     1726     1834
USTIM         17      449      513#
UUT           50#    2879
VARPAT       927#     927      934
VPATRN       882      906     1122#
WRDERR        31      865      950#     950      955      958      996
XX           129#     129      211#     211
XXT         2022#    2022     2110#    2110     2223#    2223     2266#    2266     2307#    2307
XYZ         1851#    1851
ZU            64#
$CHRIN       119
$SVUPC       206      236      246      252      272      282      288      308     2608     2721     2734     2749
$SVUUO       126      206      216      237      246      247      253      273      282      283      289      309     2609     2721
            2734     2749
$TTCHR      1989     1998     2025     2078     2086     2113     2193     2201     2226     2269     2275     2310     2315     2377
            2424     2482
$TWCNT       162     2023     2111     2224     2267     2308
$TYPNB      2122     2235
.CGOT         41     1902     1910#
.CLOSE        21      622#
.COMM         40     1884#    2138
.COMM0      1890#    1892
.COMM1      1893     1902#
.COMM2      1891     1925#
.COMM3      1932#    1939
.COMM4      1934     1940#
```

```
.COMM5      1936     1943#
.COMM6      1928     1945#
.CTBL       1915#    1929     1931
.CTYP       1885     1912#    1930
.DARG         40     2166#
.DARG0      2186#    2195
.DARG1      2182     2191#    2236
.DARGA      2187     2223#
.DARX1      2169     2181     2210     2212     2214     2220#    2229     2231     2234
.DARX2      2175     2196     2216#    2225
.EXIT         21      649#
.FHSLF       379      628      656
.ISWT         35     1392#
.ISWT2      1397#
.ISWT6      1409#
.ISWT7      1405     1413#
.ISWT8      1418#
.ISWTB      1414     1423#
.ISWTX      1428#
.ISWTY      1433     1436#
.ISWTZ      1435     1439#
.LSWIT        36     1508     1651#
.LSWTN      1502     1666#
.OARG         40     2051#
.OARG0      2071#    2080
.OARG1      2067     2076#    2123
.OARGA      2072     2110#
.OARX1      2054     2066     2095     2097     2099     2105#    2116     2118     2121
.OARX2      2060     2081     2101#    2112
.PNTOC        12      205#
.RESET        21      600#
.RSWIT        36     1563     1574     1589     1687#
.SARG         40     1964#
.SARG0      1984#    1991
.SARG1      1980     1989#    2034
.SARGA      1985     2022#
.SARX1      1967     1979     2007     2009     2011     2017#    2028     2030     2033
.SARX2      1973     1992     2013#    2024
.SLEFT      1409     1479#
.SLF1       1481#    1487
.SLF2       1486     1491#
.SLF3       1495#    1520
.SLF4       1496#    1515     1528     1532
.SLF5       1502#    1524
.SLF6       1522     1529#
.SLFX       1489     1500     1518     1534#
.SRG2       1544#    1550
.SRG3       1549     1554#
.SRG4       1557#
.SRG5       1558#    1564     1581     1600     1604
.SRG6       1566#    1587     1591     1596
.SRG6A      1588#    1590
.SRG6B      1586     1593#
```

```
.SRG7      1594     1601#
.SRGX      1552     1562      1584      1606#
.SRIGT     1423     1542#
.SWCH0     1317#    1363
.SWCH1     1318     1361#
.SWCHP       35     1307#
.SWIN      1485     1548      1619#
.SWIN0     1631#    1641
.SWPNT     1393     1444#
.SWTXT     1448     1451#
```

```
ALTCHK     162    1985    2072    2187    2266    2307
CFIBF      142
CLRBFI     144
CLZFF      629
DROPDV     624
ERCAL      381     630
FMSG       160     193     292     293     294     295     296     297     298     299     300     301     302     303
           304     305     306     307     475     487     499     533     541     543     545     581     583    1314
          1324    1325    1326    1327    1328    1329    1330    1331    1332    1333    1334    1335    1336    1337
          1338    1339    1340    1342    1343    1344    1345    1346    1347    1348    1350    1351    1352    1353
          1354    1355    1356    1359    1360    1484    1505    1507    1547    1571    1573    1734    1736    1737
          1738    1739    1740    1741    1742    1743    1744    1795    1796    1797    1798    1799    1800    1801
          1802    1803    1804    1805    1806    1807    1808    1809    1810    1811    1812    1813    1814    1815
          1816    1817    1818    1819
FMSGC      531     573     720     737    1308    1321    1400    1436    1445    1732    1754    1840    1841    2336
FMSGCD    1854    1857    2032    2120    2139    2233    2282    2322
FMSGD      552     585     726     743    1745    1759    1761    2342
FREAD      699     710
FSELEC     678
GET         95      96     101     102     125     128     129     145     146     173     174     189     198     211
           236     237     252     253     272     273     283     289     308     309     347     348     388     400
           407     456     503     504     555     556     588     589     634     664     681     690     717     728
           744     793     794     813     847     848     960     961     985    1026    1027    1178    1179    1192
          1193    1205    1206    1365    1366    1439    1440    1534    1535    1606    1607    1716    1762    1763
          1865    1866    1897    1898    1904    1905    1940    1941    1943    1944    1945    1946    2017    2018
          2022    2105    2106    2110    2144    2145    2220    2221    2223    2266    2285    2286    2307    2326
          2327    2344    2384    2385    2391    2392    2431    2432    2438    2439    2491    2492    2498    2499
          2511    2720    2721    2733    2734    2748    2749    2872    2873
GETTAB     371
GO         344     399     406     439     532     544     546     584     680     766     769     773     776     779
           786     789     804     807     810     816     833     912     913     938     939     979    1129    1130
          1138    1139    1140    1312    1315    1393    1397    1409    1418    1423    1431    1446    1485    1513
          1548    1579    1621    1642    1750    1756    1835    1838    1844    1845    1847    1855    1891    1978
          2006    2065    2094    2138    2180    2209    2263    2304    2359    2379    2389    2406    2426    2436
          2452    2484    2496    2507    2623    2629    2675    2678    2717    2742    2760    2762    2819    2871
          2885    2887
HALTF      657
JEM       1867
PBELL     2685
PCRL      1015    1966    2053    2168    2256    2300    2842    2848
PCRL2F    1493    1555    1628
PCRLF     1364    1402    1438    1482    1491    1516    1545    1582    1625    1634
PDEC      1016
PFORCE    1444    2757
PJRST     1621    1628    1633    2840    2846    2866
PNTCHF     716
PNTCI      686
PNTCIF     197     723     740    2339
PNTDCF     196     470     483     495     551
PNTDEC     187     472     485     497     956    2783    2602
PNTDS      250
PNTDSF     286
PNTHW     1019    1022    2787    2805
```

```
PNTHWF     1313    1401    1437    1714    1843
PNTMSF      160     193     292     293     294     295     296     297     298     299     300     301     302     303
            304     305     306     307     475     487     499     531     533     541     543     545     552     573
            581     583     585     720     726     737     743    1308    1314    1321    1324    1325    1326    1327
           1328    1329    1330    1331    1332    1333    1334    1335    1336    1337    1338    1339    1340    1342
           1343    1344    1345    1346    1347    1348    1350    1351    1352    1353    1354    1355    1356    1359
           1360    1400    1436    1445    1484    1492    1505    1507    1547    1554    1571    1573    1732    1734
           1736    1737    1738    1739    1740    1741    1742    1743    1744    1745    1754    1759    1761    1795
           1796    1797    1798    1799    1800    1801    1802    1803    1804    1805    1806    1807    1808    1809
           1810    1811    1812    1813    1814    1815    1816    1817    1818    1819    1840    1841    1854    1857
           2032    2120    2139    2233    2282    2322    2336    2342
PNTMSG      184     188     223     224     225     226     227     228     229     230     231     232     233     256
            257     258     259     260     261     262     263     264     265     266     267     268     269     270
            271     477     489     501     683     689     954     957    1010    1011    1017    1020    1212    1213
           1214    1215    1216    1217    1218    1219    1220    1221    1222    1223    1224    1225    1226    1227
           1228    1229    1230    1231    1232    1233    1234    1235    1236    1237    1238    1239    1240    1241
           1242    1243    1244    1245    1246    1247    1248    1249    1250    1251    1252    1253    1254    1255
           1256    1257    1258    1259    1260    1261    1262    1263    1264    1265    1266    1267    1268    1269
           1270    1271    1272    1273    1274    1275    1276    1277    1278    1279    1280    1281    1282    1283
           1284    1285    1286    1287    1288    1289    1290    1291    1292    1293    1294    1295    1296    1297
           1298    1299    1448    1510    1576    1712    1715    2679    2759    2761    2769    2773    2777    2780
           2784    2785    2792    2796    2799    2803    2833    2843    2849    2880    2881    2882    2883    2884
           2886    2891    2893    2897    2899
PNTOCF      538     542     578     582    1483    1546
PNTOCS      235    2772    2776    2795
PNTSIX      685     688
PNTSXF      536     576     722     725     739     742    1509    1575    2338    2341
PUT          78      79     118     139     140     157     158     183     192     205     206     208     245     246
            281     282     334     335     367     396     403     431     432     520     521     562     563     625
            652     676     709     736     762     763     803     830     831     877     878     965     966     993
            994    1169    1170    1183    1184    1197    1198    1307    1308    1392    1393    1479    1480    1542
           1543    1711    1723    1724    1836    1837    1884    1885    1925    1926    1964    1965    2051    2052
           2135    2136    2166    2167    2252    2253    2296    2297    2335    2354    2355    2401    2402    2448
           2449    2506    2608    2609    2610    2611    2870    2871
RESET       605
RGET         95     101     145     173     236     252     272     288     308     347     456     503     555     588
            793     847     960    1026    1178    1192    1205    1365    1439    1534    1606    1762    1897    1904
           1940    1943    1945    2017    2105    2144    2220    2285    2326    2384    2391    2431    2438    2491
           2498    2720    2733    2748    2872
RPUT         78     139     157     205     245     281     334     431     520     562     762     830     877     965
            993    1169    1183    1197    1307    1392    1479    1542    1723    1836    1884    1925    1964    2051
           2135    2166    2252    2296    2354    2401    2448    2610    2870
RTN          96     103     138     146     174     190     199     237     253     273     289     309     348     381
            339     401     408     457     504     556     589     601     606     623     630     635     650     665
            682     692     700     703     718     729     745     794     814     820     848     876     914     940
            961    1027    1154    1159    1165    1179    1193    1206    1366    1377    1440    1447    1449    1535
           1607    1626    1629    1635    1645    1717    1763    1898    1906    1942    1944    1948    2018    2106
           2145    2221    2286    2327    2334    2345    2381    2385    2392    2428    2432    2439    2486    2492
           2499    2512    2721    2736    2751    2788    2806    2815    2829    2834    2844    2847    2850    2851
           2856    2867    2869    2873    2888    2894    2900
SWITCH     1375
TIME        380
TMSG        184     188     223     224     225     226     227     228     229     230     231     232     233     256
```

```
               257     258     259     260     261     262     263     264     265     266     267     268     269     270
               271     477     489     501     957    1017    1020    1212    1213    1214    1215    1216    1217    1218
              1219    1220    1221    1222    1223    1224    1225    1226    1227    1228    1229    1230    1231    1232
              1233    1234    1235    1236    1237    1238    1239    1240    1241    1242    1243    1244    1245    1246
              1247    1248    1249    1250    1251    1252    1253    1254    1255    1256    1257    1258    1259    1260
              1261    1262    1263    1264    1265    1266    1267    1268    1269    1270    1271    1272    1273    1274
              1275    1276    1277    1278    1279    1280    1281    1282    1283    1284    1285    1286    1287    1288
              1289    1290    1291    1292    1293    1294    1295    1296    1297    1298    1299    1510    1576    1712
              2761    2773    2777    2780    2784    2785    2792    2796    2799    2803    2880    2881    2882    2883
              2884    2886    2891    2893    2897    2899
TMSGC          683     954    1010    1011    1715    2758    2769    2833
TMSGCD        2679
TMSGD          689
TTALTM        2710    2725
TTICNV        2186    2265    2306
TTIOCT        2071
TTSIXB         161    1984
TTYINP        1511    1577    1619    1639
```

```
                 LINK symbol map of      DFPTA   version 6(2364)          page 1
          Produced by LINK version 6(2364) on  1-Jul-85 at 10:17:17

          Low  segment starts at       0 ends at  240200 length   240201 = 161P
          383 words free in Low segment
          528 Global symbols loaded, therefore min. hash size is 587
          Start address is  30000, located in program DFPTA1

                 **************

JOBDAT-INITIAL-SYMBOLS
          %JOBDT       43200000454 Global  Absolute    Suppressed  .JB41                41 Global  Absolute
          .JBAPR               125 Global  Absolute                .JBBLT               45 Global  Absolute
          .JBCHN               131 Global  Absolute                .JBCNI              126 Global  Absolute
          .JBCOR               133 Global  Absolute                .JBCST              136 Global  Absolute
          .JBDA                140 Global  Absolute                .JBDDT               74 Global  Absolute
          .JBEDV               112 Global  Absolute                .JBERR               42 Global  Absolute
          .JBFF                121 Global  Absolute                .JBH41                1 Global  Absolute    Suppressed
          .JBHCR                 2 Global  Absolute    Suppressed  .JBHDA               10 Global  Absolute    Suppressed
          .JBHGA                 7 Global  Absolute    Suppressed  .JBHGH           400000 Global  Absolute    Suppressed
          .JBHNM                 5 Global  Absolute    Suppressed  .JBHRL              115 Global  Absolute
          .JBHRN                 3 Global  Absolute    Suppressed  .JBHSA                0 Global  Absolute    Suppressed
          .JBHSM                 6 Global  Absolute    Suppressed  .JBHVR                4 Global  Absolute    Suppressed
          .JBINT               134 Global  Absolute                .JBOPC              130 Global  Absolute
          .JBOPS               135 Global  Absolute                .JBOVL              131 Global  Absolute
          .JBPFH               123 Global  Absolute                .JBPFI               74 Global  Absolute
          .JBREL                44 Global  Absolute                .JBREN              124 Global  Absolute
          .JBSA                120 Global  Absolute                .JBSYM              116 Global  Absolute
          .JBTPC               127 Global  Absolute                .JBUSY              117 Global  Absolute
          .JBUUO                40 Global  Absolute                .JBVER              137 Global  Absolute

                 **************

DFPTA     from DSK:DFPTAT.REL     created by MACRO on  1-Jul-85 at  7:46:39
          Low  segment starts at     140 ends at   30740 length    30601 (octal), 12673. (decimal)


                 **************

DFPTA1    from DSK:DFPTA1.REL     created by MACRO on  1-Jul-85 at  7:48:28
          Low  segment starts at   30741 ends at  115621 length    64661 (octal), 27057. (decimal)

          AAPNT              44732 Entry   Absolute                AEBUF             45141 Global  Absolute
          AEXEC              44617 Entry   Absolute                ALACT             44725 Global  Absolute
          ALCOR              44724 Global  Absolute                ALCSR             44721 Global  Absolute
          ALEBF              44722 Global  Absolute                ALERR             44723 Global  Absolute
          ALFLC              44730 Global  Absolute                ALFLE             44727 Global  Absolute
          ALFLS              44726 Global  Absolute                ALFLT             44731 Global  Absolute
          ALSRT              44720 Global  Absolute                ALTF             115531 Global  Relocatable
          ALTIM              44717 Global  Absolute                BBPNT             45144 Entry   Absolute
          BEXEC              44770 Entry   Absolute                BUFF              50000 Global  Absolute
          CADDR              37750 Global  Absolute                CALPAR            43625 Entry   Absolute
          CBUF               52003 Entry   Absolute                CCPNT             46044 Entry   Absolute
          CCSR               46027 Global  Absolute                CDERR             46034 Global  Absolute
          CEBUF              45142 Global  Absolute                CEBUFA            46032 Global  Absolute
          CEBUFC             46031 Global  Absolute                CEXEC             45512 Entry   Absolute
```

```
                LINK symbol map of      DFPTA   version 6(2364)         page 2
DFPTA1                                                                                                   SEQ 2164
        CISEL             47004 Entry   Absolute                CLEN              46036 Global  Absolute
        CPAT              46035 Global  Absolute                CSRENG            43737 Entry   Absolute
        CSRPNT            43715 Entry   Absolute                CSTATF            46040 Global  Absolute
        CSTWRD            52320 Entry   Absolute                CWORDL            37751 Global  Absolute
        CWORDR            37752 Global  Absolute                DISMS        104000000167 Global  Absolute
        DRCRAM            40430 Entry   Absolute                DWCRAM            40376 Entry   Absolute
        FINECH           115543 Global  Relocatable             FINPUT           115544 Global  Relocatable
        GETJI      104000000507 Global  Absolute                GTAD         104000000227 Global  Absolute
        IADDR             46570 Global  Absolute                ICSR              46572 Global  Absolute
        IDEBUG           115547 Global  Relocatable             IEXEC             46146 Entry   Absolute
        IFLAG             46574 Global  Absolute                IIOPF             46575 Global  Absolute
        IIPNT             46610 Entry   Absolute                INDLIS            31275 Entry   Absolute
        IPRINT            42640 Entry   Absolute                LDEBUG           115551 Global  Relocatable
        MDEBUG           115555 Global  Relocatable             MLOADN            43207 Entry   Absolute
        MLOADY            43211 Entry   Absolute                NISEL             47003 Entry   Absolute
        NSSTEP            44550 Global  Absolute                ODELAY            43121 Entry   Absolute
        PARFLG            35764 Global  Absolute                PGMNAM            30600 Global  Absolute
        PMODE             47034 Global  Absolute                PORSEL            47005 Entry   Absolute
        PORTCI            47002 Entry   Absolute                PORTNI            47001 Entry   Absolute
        QADDR             46025 Global  Absolute                QADDRF            46026 Global  Absolute
        SAADDR            44553 Global  Absolute                SDEBUG           115602 Global  Relocatable
        SEADDR            44552 Global  Absolute                SEXEC             44416 Entry   Absolute
        SPEAR2           115605 Entry   Relocatable             SPREP2            44234 Entry   Absolute
        SSADDR            44551 Global  Absolute                SSPNT             44554 Entry   Absolute
        START             30610 Entry   Absolute                STARTA            30710 Entry   Absolute
        SYERR      104000000527 Global  Absolute                TADDR             45430 Global  Absolute
        TADDRF            45431 Global  Absolute                TAKFIL            32747 Global  Absolute
        TCSRF             45432 Global  Absolute                TEBUFA            45435 Global  Absolute
        TEBUFC            45434 Global  Absolute                TEXEC             45227 Entry   Absolute
        TLOAD             43661 Entry   Absolute                TPAT              52002 Global  Absolute
        TRACE             42717 Entry   Absolute                TSLOD1           115607 Global  Relocatable
        TSLOD2           115610 Global  Relocatable             TSTADD            47015 Entry   Absolute
        TSTCLS            47010 Entry   Absolute                TSTFLG            47014 Entry   Absolute
        TSTINH            47023 Global  Absolute                TSTMIC            47016 Entry   Absolute
        TSTMSK            47017 Entry   Absolute                TSTNAM            47007 Entry   Absolute
        TSTNUM            47011 Entry   Absolute                TSTOFF            47026 Entry   Absolute
        TSTPC             47013 Entry   Absolute                TSTPNT            43076 Entry   Absolute
        TSTREP            47020 Entry   Absolute                TSTSUB            47012 Entry   Absolute
        TTPNT             45440 Entry   Absolute                TWORD             52000 Global  Absolute
        UDEBUG           115615 Global  Relocatable             UUT               47006 Entry   Absolute

                **************

DFPTA2  from DSK:DFPTA2.REL     created by MACRO on  1-Jul-85 at  7:53:55
        Low  segment starts at  115622 ends at  131516 length    13675 (octal),  6077. (decimal)

        TSTE1            115622 Entry   Relocatable             TSTE10           116440 Entry   Relocatable
        TSTE11           116526 Entry   Relocatable             TSTE12           116557 Entry   Relocatable
        TSTE13           116615 Entry   Relocatable             TSTE14           116730 Entry   Relocatable
        TSTE15           117004 Entry   Relocatable             TSTE16           117042 Entry   Relocatable
        TSTE17           117220 Entry   Relocatable             TSTE2            115647 Entry   Relocatable
        TSTE20           117246 Entry   Relocatable             TSTE21           117275 Entry   Relocatable
        TSTE22           117330 Entry   Relocatable             TSTE23           117406 Entry   Relocatable
        TSTE24           117615 Entry   Relocatable             TSTE25           117665 Entry   Relocatable
```

```
               LINK symbol map of      DFPTA   version 6(2364)        page 3
DFPTA2
    TSTE26            117755 Entry   Relocatable       TSTE27            120033 Entry   Relocatable
    TSTE3             115745 Entry   Relocatable       TSTE30            120114 Entry   Relocatable
    TSTE31            120171 Entry   Relocatable       TSTE32            120251 Entry   Relocatable
    TSTE33            120323 Entry   Relocatable       TSTE34            120376 Entry   Relocatable
    TSTE35            120466 Entry   Relocatable       TSTE36            120760 Entry   Relocatable
    TSTE37            121034 Entry   Relocatable       TSTE4             116036 Entry   Relocatable
    TSTE40            121063 Entry   Relocatable       TSTE41            121175 Entry   Relocatable
    TSTE42            121342 Entry   Relocatable       TSTE43            121414 Entry   Relocatable
    TSTE44            121465 Entry   Relocatable       TSTE45            121535 Entry   Relocatable
    TSTE46            121604 Entry   Relocatable       TSTE47            121655 Entry   Relocatable
    TSTE5             116131 Entry   Relocatable       TSTE50            122055 Entry   Relocatable
    TSTE51            122254 Entry   Relocatable       TSTE52            122452 Entry   Relocatable
    TSTE53            122457 Entry   Relocatable       TSTE54            122526 Entry   Relocatable
    TSTE55            122573 Entry   Relocatable       TSTE56            122647 Entry   Relocatable
    TSTE57            122720 Entry   Relocatable       TSTE6             116220 Entry   Relocatable
    TSTE60            122765 Entry   Relocatable       TSTE61            123142 Entry   Relocatable
    TSTE62            123223 Entry   Relocatable       TSTE63            123325 Entry   Relocatable
    TSTE64            123425 Entry   Relocatable       TSTE65            123652 Entry   Relocatable
    TSTE66            124076 Entry   Relocatable       TSTE67            124103 Entry   Relocatable
    TSTE7             116254 Entry   Relocatable       TSTE70            124110 Entry   Relocatable
    TSTE71            124344 Entry   Relocatable       TSTE72            124606 Entry   Relocatable
    TSTE73            125041 Entry   Relocatable       TSTE74            125224 Entry   Relocatable
    TSTE75            125435 Entry   Relocatable       TSTE76            125566 Entry   Relocatable

               **************

DFPTA3  from DSK:DFPTA3.REL     created by MACRO on  1-Jul-85 at  8:05:42
        Low  segment starts at  131517 ends at  143734 length    12216 (octal),  5262. (decimal)

    TSTS1             131517 Entry   Relocatable       TSTS10            132431 Entry   Relocatable
    TSTS11            132504 Entry   Relocatable       TSTS12            132552 Entry   Relocatable
    TSTS13            132632 Entry   Relocatable       TSTS14            132702 Entry   Relocatable
    TSTS15            133006 Entry   Relocatable       TSTS16            133200 Entry   Relocatable
    TSTS17            133372 Entry   Relocatable       TSTS2             131660 Entry   Relocatable
    TSTS20            133566 Entry   Relocatable       TSTS21            133756 Entry   Relocatable
    TSTS22            134152 Entry   Relocatable       TSTS23            134230 Entry   Relocatable
    TSTS24            134310 Entry   Relocatable       TSTS25            134372 Entry   Relocatable
    TSTS26            134456 Entry   Relocatable       TSTS27            134544 Entry   Relocatable
    TSTS3             131745 Entry   Relocatable       TSTS30            134551 Entry   Relocatable
    TSTS31            134641 Entry   Relocatable       TSTS32            134755 Entry   Relocatable
    TSTS33            135033 Entry   Relocatable       TSTS34            135077 Entry   Relocatable
    TSTS35            135163 Entry   Relocatable       TSTS36            135425 Entry   Relocatable
    TSTS37            135517 Entry   Relocatable       TSTS4             132046 Entry   Relocatable
    TSTS40            135615 Entry   Relocatable       TSTS41            135715 Entry   Relocatable
    TSTS42            135775 Entry   Relocatable       TSTS43            136074 Entry   Relocatable
    TSTS44            136166 Entry   Relocatable       TSTS45            136246 Entry   Relocatable
    TSTS46            136335 Entry   Relocatable       TSTS47            136431 Entry   Relocatable
    TSTS5             132147 Entry   Relocatable       TSTS50            136521 Entry   Relocatable
    TSTS51            137105 Entry   Relocatable       TSTS52            137175 Entry   Relocatable
    TSTS53            137372 Entry   Relocatable       TSTS54            140014 Entry   Relocatable
    TSTS55            140116 Entry   Relocatable       TSTS56            140426 Entry   Relocatable
    TSTS57            140516 Entry   Relocatable       TSTS6             132303 Entry   Relocatable
    TSTS60            140570 Entry   Relocatable       TSTS61            140646 Entry   Relocatable
    TSTS62            140742 Entry   Relocatable       TSTS63            141042 Entry   Relocatable
```

```
                LINK symbol map of      DFPTA   version 6(2364)         page 4
DFPTA3
        TSTS64          141112 Entry    Relocatable             TSTS65          141262 Entry   Relocatable
        TSTS66          141427 Entry    Relocatable             TSTS67          141633 Entry   Relocatable
        TSTS7           132356 Entry    Relocatable             TSTS70          142012 Entry   Relocatable

                **************

DFPTA4  from DSK:DFPTA4.REL     created by MACRO on  1-Jul-85 at  8:24:43
        Low  segment starts at  143735 ends at  153760 length    10024 (octal),  4116. (decimal)

        TSTA1           143735 Entry    Relocatable             TSTA10          145147 Entry   Relocatable
        TSTA11          145231 Entry    Relocatable             TSTA12          145313 Entry   Relocatable
        TSTA13          145375 Entry    Relocatable             TSTA14          146034 Entry   Relocatable
        TSTA15          146115 Entry    Relocatable             TSTA16          146331 Entry   Relocatable
        TSTA17          146610 Entry    Relocatable             TSTA2           144210 Entry   Relocatable
        TSTA20          147065 Entry    Relocatable             TSTA21          147356 Entry   Relocatable
        TSTA22          147721 Entry    Relocatable             TSTA23          150402 Entry   Relocatable
        TSTA24          150707 Entry    Relocatable             TSTA25          151216 Entry   Relocatable
        TSTA26          151507 Entry    Relocatable             TSTA27          151752 Entry   Relocatable
        TSTA3           144573 Entry    Relocatable             TSTA30          152243 Entry   Relocatable
        TSTA4           144655 Entry    Relocatable             TSTA5           144730 Entry   Relocatable
        TSTA6           145003 Entry    Relocatable             TSTA7           145065 Entry   Relocatable

                **************

DFPTA5  from DSK:DFPTA5.REL     created by MACRO on  1-Jul-85 at  8:41:00
        Low  segment starts at  153761 ends at  157672 length     3712 (octal),  1994. (decimal)

        TSTA31          153761 Entry    Relocatable
                **************

DFPTA6  from DSK:DFPTA6.REL     created by MACRO on  1-Jul-85 at  9:04:27
        Low  segment starts at  157673 ends at  164254 length     4362 (octal),  2290. (decimal)

        TSTA32          157673 Entry    Relocatable

                **************

DFPTA7  from DSK:DFPTA7.REL     created by MACRO on  1-Jul-85 at  9:17:53
        Low  segment starts at  164255 ends at  170152 length     3676 (octal),  1982. (decimal)

        TSTA33          164255 Entry    Relocatable             TSTA34          164412 Entry   Relocatable
        TSTA35          164554 Entry    Relocatable             TSTA36          165610 Entry   Relocatable
        TSTA37          166273 Entry    Relocatable             TSTA40          166521 Entry   Relocatable
        TSTA41          166744 Entry    Relocatable             TSTA42          167164 Entry   Relocatable

                **************

DFPTA8  from DSK:DFPTA8.REL     created by MACRO on  1-Jul-85 at  9:39:59
        Low  segment starts at  170153 ends at  203575 length    13423 (octal),  5907. (decimal)

        TSTU1           170153 Entry    Relocatable             TSTU10          171500 Entry   Relocatable
        TSTU11          171604 Entry    Relocatable             TSTU12          171662 Entry   Relocatable
        TSTU13          172016 Entry    Relocatable             TSTU14          172106 Entry   Relocatable
        TSTU15          172404 Entry    Relocatable             TSTU16          172642 Entry   Relocatable
```

```
               LINK symbol map of      DFPTA   version 6(2364)         page 5
DFPTA8
        TSTU17          173470 Entry   Relocatable              TSTU2           170327 Entry   Relocatable
        TSTU20          173700 Entry   Relocatable              TSTU21          174052 Entry   Relocatable
        TSTU22          174243 Entry   Relocatable              TSTU23          174434 Entry   Relocatable
        TSTU24          174625 Entry   Relocatable              TSTU25          175020 Entry   Relocatable
        TSTU26          175210 Entry   Relocatable              TSTU27          175371 Entry   Relocatable
        TSTU3           170403 Entry   Relocatable              TSTU30          175665 Entry   Relocatable
        TSTU31          176122 Entry   Relocatable              TSTU32          176462 Entry   Relocatable
        TSTU33          176756 Entry   Relocatable              TSTU34          177200 Entry   Relocatable
        TSTU35          177312 Entry   Relocatable              TSTU36          200501 Entry   Relocatable
        TSTU37          200674 Entry   Relocatable              TSTU4           170460 Entry   Relocatable
        TSTU40          200762 Entry   Relocatable              TSTU41          201044 Entry   Relocatable
        TSTU5           170557 Entry   Relocatable              TSTU6           170663 Entry   Relocatable
        TSTU7           171172 Entry   Relocatable
           **************

DFPTA9  from DSK:DFPTA9.REL     created by MACRO on  1-Jul-85 at 10:01:43
        Low  segment starts at  203576 ends at  217662 length    14065 (octal),  6197. (decimal)

        TSTC1           203576 Entry   Relocatable
        TSTC10          206315 Entry   Relocatable              TSTC11          206632 Entry   Relocatable
        TSTC12          206677 Entry   Relocatable              TSTC13          207006 Entry   Relocatable
        TSTC14          207306 Entry   Relocatable              TSTC15          207605 Entry   Relocatable
        TSTC16          210120 Entry   Relocatable              TSTC17          210431 Entry   Relocatable
        TSTC2           204435 Entry   Relocatable              TSTC20          210742 Entry   Relocatable
        TSTC21          211254 Entry   Relocatable              TSTC22          211534 Entry   Relocatable
        TSTC23          212013 Entry   Relocatable              TSTC24          212271 Entry   Relocatable
        TSTC25          212552 Entry   Relocatable              TSTC26          212660 Entry   Relocatable
        TSTC27          213170 Entry   Relocatable              TSTC3           204642 Entry   Relocatable
        TSTC30          213506 Entry   Relocatable              TSTC31          214022 Entry   Relocatable
        TSTC32          214340 Entry   Relocatable              TSTC33          214656 Entry   Relocatable
        TSTC34          215142 Entry   Relocatable              TSTC35          215425 Entry   Relocatable
        TSTC36          215707 Entry   Relocatable              TSTC37          216170 Entry   Relocatable
        TSTC4           204750 Entry   Relocatable              TSTC5           205073 Entry   Relocatable
        TSTC6           205410 Entry   Relocatable              TSTC7           206013 Entry   Relocatable

           **************

DFPTAA  from DSK:DFPTAA.REL     created by MACRO on  1-Jul-85 at 10:14:07
        Low  segment starts at  217663 ends at  225613 length     5731 (octal),  3033. (decimal)

        TSTC40          217663 Entry   Relocatable              TSTC41          220046 Entry   Relocatable
        TSTC42          220276 Entry   Relocatable              TSTC43          220366 Entry   Relocatable
        TSTC44          220455 Entry   Relocatable              TSTC45          220670 Entry   Relocatable
        TSTC46          221055 Entry   Relocatable              TSTC47          221211 Entry   Relocatable
        TSTC50          221436 Entry   Relocatable              TSTC51          221553 Entry   Relocatable
        TSTC52          222024 Entry   Relocatable              TSTC53          222076 Entry   Relocatable
        TSTC54          222463 Entry   Relocatable              TSTC55          223033 Entry   Relocatable
        TSTC56          223354 Entry   Relocatable              TSTC57          223676 Entry   Relocatable
        TSTC60          224235 Entry   Relocatable
           **************
```

```
                 LINK symbol map of      DFPTA   version 6(2364)        page 6

DFPTAI  from DSK:DFPTAI.REL     created by MACRO on  1-Jul-85 at 10:15:46
        Low  segment starts at  225614 ends at  232036 length     4223 (octal),  2195. (decimal)

        AADDR         226237 Global  Relocatable
        CBASE         230111 Global  Relocatable                CCWPNT        227671 Entry   Relocatable
        CHDATA        230205 Entry   Relocatable                CHINIT        227262 Entry   Relocatable
        CHKCSR        226132 Entry   Relocatable                CLREBU        226116 Entry   Relocatable
        DEVREL        230314 Entry   Relocatable                DEVREQ        230510 Entry   Relocatable
        DSETUP        230222 Entry   Relocatable                EADDR         226236 Global  Relocatable
        EPCAP%  104000000151 Global  Absolute                   ERESET        225774 Entry   Relocatable
        ERJMP   320700000000 Global  Absolute                   ERSTR%  104000000011 Global  Absolute
        FFF           230754 Entry   Relocatable                GENCCW        227320 Entry   Relocatable
        GETLOG        227470 Entry   Relocatable                HAVECI        230506 Global  Relocatable
        HAVENI        230507 Global  Relocatable                IDELAY        226701 Global  Relocatable
        INITPD        226346 Entry   Relocatable                INITPI        226322 Entry   Relocatable
        INTAPR        226777 Global  Relocatable                INTCON        226753 Global  Relocatable
        INTCSR        227073 Global  Relocatable                INTEND        227143 Global  Relocatable
        INTNUM        226702 Global  Relocatable                INTPC         226703 Global  Relocatable
        INTTIM        227023 Global  Relocatable                INTTYP        226727 Global  Relocatable
        INTUSE        227047 Global  Relocatable                IPACLR        226057 Entry   Relocatable
        IPASRT        226151 Entry   Relocatable                IPASST        226240 Entry   Relocatable
        IPASTP        226203 Entry   Relocatable                ISTOP         226003 Entry   Relocatable
        LDCRAM        225651 Entry   Relocatable                LDCSR         225624 Entry   Relocatable
        LDEBUF        225614 Entry   Relocatable                LDRAR         225635 Entry   Relocatable
        LOGBUF        230133 Global  Relocatable                LOGPNT        227614 Entry   Relocatable
        NI%     104000000630 Global  Absolute                   PF500         227220 Global  Relocatable
        PF501         227221 Global  Relocatable                PF502         227222 Global  Relocatable
        PFAIL         227224 Global  Relocatable                PFAPR         227217 Global  Relocatable
        PFCLR         227225 Entry   Relocatable                PFSET         227175 Entry   Relocatable
        PISYOF        226400 Entry   Relocatable                PISYON        226403 Entry   Relocatable
        RDCRAM        225722 Entry   Relocatable                RDCSR         225674 Entry   Relocatable
        RDEBUF        225664 Entry   Relocatable                RDLAR         225711 Entry   Relocatable
        REL           230370 Entry   Relocatable                REQ1          230562 Entry   Relocatable
        REQ2          230622 Entry   Relocatable                RPCAP%  104000000150 Global  Absolute
        SDATA         226235 Global  Relocatable                SETEBU        226064 Entry   Relocatable
        SETLAR        226101 Entry   Relocatable                SETVEC        226406 Entry   Relocatable
        SLAST         226234 Global  Relocatable                SNEXT         226233 Global  Relocatable
        USRIO%  104000000310 Global  Absolute                   USRIOF        230730 Entry   Relocatable
        USRION        230711 Entry   Relocatable                .CONI         225733 Entry   Relocatable
        .CONO         225743 Entry   Relocatable                .DATAI        225754 Entry   Relocatable
        .DATAO        225764 Entry   Relocatable                .EICHN             2 Global  Absolute     Suppressed
        .EIFCN             0 Global  Absolute     Suppressed    .EIGET            20 Global  Absolute     Suppressed
        .EIRCI            15 Global  Absolute     Suppressed    .EIREL            21 Global  Absolute     Suppressed
        .EISOF            11 Global  Absolute     Suppressed    .EISTA             4 Global  Absolute     Suppressed
        .FHSLF        400000 Global  Absolute     Suppressed    .INPNT        227144 Entry   Relocatable
        .INWD1        227173 Global  Relocatable                .INWD2        227174 Global  Relocatable
        .PIOFF        226352 Entry   Relocatable                .PION         226364 Entry   Relocatable
        .PRIOU           101 Global  Absolute     Suppressed
                ***************
```

```
             LINK symbol map of      DFPTA   version 6(2364)        page 7

DFPTAU  from DSK:DFPTAU.REL     created by MACRO on  1-Jul-85 at 10:16:37
        Low  segment starts at  232037 ends at  240200 length     6142 (octal),  3170. (decimal)

        ALTCHH            232070 Entry   Relocatable
        ARGFLG            235336 Global  Relocatable          ARGUM             235335 Global  Relocatable
        BUFCOM            233034 Entry   Relocatable          BUFGEN            233045 Entry   Relocatable
        CHKARG            235337 Entry   Relocatable          CLRBUF            232104 Entry   Relocatable
        CLZFF       104000000034 Global  Absolute             CONVSX            232037 Entry   Relocatable
        CURTIM            232456 Global  Relocatable          DEBTIM            232460 Global  Relocatable
        DECYN             235241 Entry   Relocatable          ERCAL       320740000000 Global  Absolute
        FGETW             232720 Entry   Relocatable          FINCMD            235425 Entry   Relocatable
        FINIT             232635 Entry   Relocatable          FIOFF             235410 Entry   Relocatable
        FOARG             235527 Entry   Relocatable          FORPNT            240160 Global  Relocatable
        HALTF       104000000170 Global  Absolute             LASARG            235363 Entry   Relocatable
        LCHAR             240165 Global  Relocatable          MAXTIM            232457 Global  Relocatable
        MONTIM            232321 Entry   Relocatable          PARSER            234756 Entry   Relocatable
        PATBUF            233231 Entry   Relocatable          PATPNT            233441 Entry   Relocatable
        PNTDEV            236104 Entry   Relocatable          PPDEC             232222 Entry   Relocatable
        PPDECF            232250 Entry   Relocatable          PSDN              232144 Entry   Relocatable
        PSTAMP            232462 Entry   Relocatable          PTIME             232350 Entry   Relocatable
        RESET       104000000147 Global  Absolute             RUNTME            232364 Entry   Relocatable
        SCOOFF            236121 Global  Relocatable          SCOPE1            235620 Entry   Relocatable
        SCOSW             240172 Global  Relocatable          STCLOK            232276 Entry   Relocatable
        SWCHPT            234626 Entry   Relocatable          SWCOM             234635 Entry   Relocatable
        SWITT             233630 Entry   Relocatable          SWRGT             233635 Global  Relocatable
        TIME        104000000014 Global  Absolute             TTYYES            232120 Entry   Relocatable
        UNAME             232716 Global  Relocatable          USTIM             232461 Global  Relocatable
        WRDERR            240175 Global  Relocatable          .CGOT             235044 Global  Relocatable
        .CLOSE            232602 Entry   Relocatable          .COMM             235020 Entry   Relocatable
        .DARG             235262 Entry   Relocatable          .EXIT             232620 Entry   Relocatable
        .ISWT             233636 Entry   Relocatable          .LSWIT            234550 Global  Relocatable
        .OARG             235166 Entry   Relocatable          .PNTOC            232160 Entry   Relocatable
        .RESET            232573 Entry   Relocatable          .RSWIT            234604 Global  Relocatable
        .SARG             235115 Entry   Relocatable          .SWCHP            233571 Entry   Relocatable

             ***************
```

```
Index to LINK symbol map of     DFPTA   version 6(2364)        page 8

Name      Page      Name      Page      Name      Page      Name      Page

DFPTA      1        DFPTA4     4        DFPTA7     4        DFPTAA     5
DFPTA1     1        DFPTA5     4        DFPTA8     4        DFPTAI     6
DFPTA2     2        DFPTA6     4        DFPTA9     5        DFPTAU     7
DFPTA3     3

[End of LINK map of     DFPTA]
```

2158